
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTTTTTTTTTTTTTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ OOO OOO BBBBBBBBBBBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJ JJJ OOO OOO BBB BBB CCC TTT LLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL
JJJJJJJJJ OOOOOOOOO BBBBBBBBBBBB CCCCCCCCCCCC TTT LLLLLLLLLLLLLLL

FILEID**QUEUEUTIL

```
;     1          0001  0 MODULE QUEUEUTIL(%TITLE 'Queue manipulation utilities'
;     2          0002  0                   IDENT = 'V04-000'
;     3          0003  0                   ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !**********************************************************************************
;     8          0008  1 !*                                                                                *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                           *
;    12          0012  1 !*                                                                                *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED         *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
;    18          0018  1 !*  TRANSFERRED.                                                                   *
;    19          0019  1 !*                                                                                *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE         *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT        *
;    22          0022  1 !*  CORPORATION.                                                                   *
;    23          0023  1 !*                                                                                *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;    26          0026  1 !*                                                                                *
;    27          0027  1 !*                                                                                *
;    28          0028  1 !**********************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !       Job controller.
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !       This module contains utility routines to manipulate the job queue.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !       VAX/VMS user and kernel mode.
;    40          0040  1 !--
;    41          0041  1 !
;    42          0042  1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
;    43          0043  1 !
;    44          0044  1 ! MODIFIED BY:
;    45          0045  1 !
;    46          0046  1 !       V03-007 KPL0002         P Lieberwirth,  23-Jul-1984
;    47          0047  1 !               Protect routine DEQUEUE_OPEN_JOB as in V03-006.
;    48          0048  1 !
;    49          0049  1 !       V03-006 KPL0001         P Lieberwirth,   9-Jul-1984
;    50          0050  1 !               Protect routine COMPLETE_JOB against common form of queue file
;    51          0051  1 !               corruption - specifically an invalid SJH.
;    52          0052  1 !
;    53          0053  1 !       V03-005 MLJ0115         Martin L. Jack, 30-Jul-1983  14:55
;    54          0054  1 !               Changes for job controller baselevel.
;    55          0055  1 !
;    56          0056  1 !       V03-004 MLJ0114         Martin L. Jack, 23-Jun-1983  5:02
;    57          0057  1 !               Changes for job controller baselevel and divide with RECORDUTL.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-003 MLJ0113         Martin L. Jack, 26-May-1983  21:08
;    60          0060  1 !               Changes for job controller baselevel.
;    61          0061  1 !
;    62          0062  1 !       V03-002 MLJ0112         Martin L. Jack, 29-Apr-1983  3:04
;    63          0063  1 !               Changes for job controller baselevel.
;    64          0064  1 !
;    65          0065  1 !       V03-001 MLJ0109         Martin L. Jack, 14-Apr-1983  12:47
;    66          0066  1 !               Changes for job controller baselevel.
;    67          0067  1 !
;    68          0068  1 !**
```

```
;    70          0069  1 REQUIRE 'SRC$:JOBCTLDEF';
;    71          1110  1
;    72          1111  1
;    73          1112  1 FORWARD ROUTINE
;    74          1113  1         ENTER_PROCESS_DATA:             NOVALUE,
;    75          1114  1         FIND_PROCESS_DATA:              L_OUTPUT_3,
;    76          1115  1         SEARCH_QUEUES:                  L_OUTPUT_4,
;    77          1116  1         DEQUEUE_OPEN_JOB:               L_OUTPUT_1,
;    78          1117  1         ALLOCATE_ENTRY_NUMBER,
;    79          1118  1         DEALLOCATE_ENTRY_NUMBER:        NOVALUE,
;    80          1119  1         JOB_STATUS_MESSAGE,
;    81          1120  1         NOTIFY_USER:                    NOVALUE,
;    82          1121  1         COMPLETE_JOB:                   NOVALUE,
;    83          1122  1         VALIDATE_OBJECT_NAME,
;    84          1123  1         FIND_CHARACTERISTIC:            L_OUTPUT_1,
;    85          1124  1         FIND_FORM_NAME:                 L_OUTPUT_2,
;    86          1125  1         FIND_FORM_NUMBER:               L_OUTPUT_2,
;    87          1126  1         FIND_QUEUE:                     L_OUTPUT_4,
;    88          1127  1         FIND_FORM_REFERENCES_J,
;    89          1128  1         FIND_FORM_REFERENCES,
;    90          1129  1         FIND_QUEUE_REFERENCES_J,
;    91          1130  1         FIND_QUEUE_REFERENCES,
;    92          1131  1         DEALLOCATE_VARIABLE_DATA:       NOVALUE,
;    93          1132  1         FETCH_VARIABLE_ITEM,
;    94          1133  1         FETCH_VARIABLE_ITEM_LIST,
;    95          1134  1         FETCH_VARIABLE_DATA:            NOVALUE,
;    96          1135  1         STORE_VARIABLE_DATA,
;    97          1136  1         STORE_VARIABLE_DATA_LIST;
;    98          1137  1
;    99          1138  1
;   100          1139  1 EXTERNAL ROUTINE
;   101          1140  1         AFTER_AST:                      NOVALUE,
;   102          1141  1         ALLOCATE_MEMORY,
;   103          1142  1         ALLOCATE_RECORD:                L_OUTPUT_2,
;   104          1143  1         BROADCAST_MESSAGE:              NOVALUE,
;   105          1144  1         DEALLOCATE_RECORD_LIST:         NOVALUE,
;   106          1145  1         DELETE_SJH_RECORD:              NOVALUE,
;   107          1146  1         ENQUEUE_JOB:                    L_OUTPUT_2 NOVALUE,
;   108          1147  1         READ_RECORD,
;   109          1148  1         RELEASE_RECORD:                 NOVALUE,
;   110          1149  1         REWRITE_RECORD:                 NOVALUE,
;   111          1150  1         SCAN_INCOMPLETE_SERVICES:       NOVALUE,
;   112          1151  1         UPDATE_GETQUI_DATA:             NOVALUE,
;   113          1152  1         WRITE_ACCOUNTING_RECORD:        NOVALUE;
;   114          1153  1
;   115          1154  1
;   116          1155  1 BUILTIN
;   117          1156  1         EDIV,
;   118          1157  1         MOVC3,
;   119          1158  1         TESTBITCS;
```

```
;  121          1159  1 GLOBAL ROUTINE ENTER_PROCESS_DATA(TYPE,PID,P1,P2): NOVALUE=
;  122          1160  1
;  123          1161  1 !++
;  124          1162  1 !
;  125          1163  1 ! FUNCTIONAL DESCRIPTION:
;  126          1164  1 !       This routine adds an entry to the process data structure.
;  127          1165  1 !
;  128          1166  1 ! INPUT PARAMETERS:
;  129          1167  1 !       TYPE            - Process type.
;  130          1168  1 !       PID             - Process ID.
;  131          1169  1 !       P1              - (Optional) First parameter.
;  132          1170  1 !       P2              - (Optional) Second parameter.
;  133          1171  1 !
;  134          1172  1 ! IMPLICIT INPUTS:
;  135          1173  1 !       NONE
;  136          1174  1 !
;  137          1175  1 ! OUTPUT PARAMETERS:
;  138          1176  1 !       NONE
;  139          1177  1 !
;  140          1178  1 ! IMPLICIT OUTPUTS:
;  141          1179  1 !       NONE
;  142          1180  1 !
;  143          1181  1 ! ROUTINE VALUE:
;  144          1182  1 !       NONE
;  145          1183  1 !
;  146          1184  1 ! SIDE EFFECTS:
;  147          1185  1 !       NONE
;  148          1186  1 !
;  149          1187  1 !--
;  150          1188  1
;  151          1189  2 BEGIN
;  152          1190  2 LOCAL
;  153          1191  2         PDB:            REF BBLOCK,     ! Pointer to PDB
;  154          1192  2         PDE:            REF BBLOCK;     ! Pointer to PDB entry
;  155          1193  2 BUILTIN
;  156          1194  2         ACTUALCOUNT;
;  157          1195  2
;  158          1196  2
;  159          1197  2 ! Search for an unused entry within the existing PDB list.
;  160          1198  2 !
;  161          1199  2 PDB = .PROCESS_DATA_LIST;
;  162          1200  2 WHILE .PDB NEQ 0 DO
;  163          1201  3     BEGIN
;  164          1202  3     IF .PDB[PDB_COUNT] LSSU PDB_K_MAX
;  165          1203  3     THEN
;  166          1204  4         BEGIN
;  167          1205  4         PDE = PDB[PDB_ENTRIES] + .PDB[PDB_COUNT] * PDE_S_ENTRY;
;  168          1206  4         EXITLOOP;
;  169          1207  3         END;
;  170          1208  3     PDB = .PDB[PDB_LINK];
;  171          1209  2     END;
;  172          1210  2
;  173          1211  2
;  174          1212  2 ! If no free entry found, allocate and initialize a new page.
;  175          1213  2 !
;  176          1214  2 IF .PDB EQL 0
;  177          1215  2 THEN
```

```
;  178          1216  3     BEGIN
;  179          1217  3     PDB = ALLOCATE_MEMORY();
;  180          1218  3     PDB[PDB_LINK] = .PROCESS_DATA_LIST;
;  181          1219  3     PROCESS_DATA_LIST = .PDB;
;  182          1220  3     PDE = PDB[PDB_ENTRIES];
;  183          1221  2     END;
;  184          1222  2
;  185          1223  2
;  186          1224  2 ! Initialize the PDB entry.
;  187          1225  2 !
;  188          1226  2 PDB[PDB_COUNT] = .PDB[PDB_COUNT] + 1;
;  189          1227  2 PDE[PDE_TYPE] = .TYPE;
;  190          1228  2 PDE[PDE_PID] = .PID;
;  191          1229  2 IF ACTUALCOUNT() GEQU 3 THEN PDE[PDE_P1] = .P1;
;  192          1230  2 IF ACTUALCOUNT() GEQU 4 THEN PDE[PDE_P2] = .P2;
;  193          1231  1 END;


                                                .TITLE  QUEUEUTIL Queue manipulation utilities
                                                .IDENT  \V04-000\

                                                .PSECT  COMMON,NOEXE,  OVR,2

                                    00000 DIAG_STORAGE_BASE:
                                                .BLKB   0
                                    00000 DIAG_TRACE:
                                                .BLKB   96
                                    00060 DIAG_COUNT:
                                                .BLKB   96
                                    000C0 DIAG_FLAGS:
                                                .BLKB   4
                                    000C4 WORK_AREA:
                                                .BLKB   44
                                    000F0 SNDJBC_COUNT:
                                                .BLKB   132
                                    00174 GETQUI_COUNT:
                                                .BLKB   40
                                    0019C SNDACC_COUNT:
                                                .BLKB   28
                                    001B8 SNDSMB_COUNT:
                                                .BLKB   72
                                    00200 DIAG_STORAGE_END:
                                                .BLKB   0
                                    00200 FLAGS:    .BLKB   4
                                    00204 IMAGE_DUMP_STSFLG:
                                                .BLKB   4
                                    00208 THIS_SYSID:
                                                .BLKB   6
                                    0020E       .BLKB   2
                                    00210 CUR_TIME:
                                                .BLKB   8
                                    00218 HOURLY_TIME:
                                                .BLKB   8
                                    00220 HOURLY_PARAMS:
                                                .BLKB   20
                                    00234 SYMBIONT_COUNT:
                                                .BLKB   4
```

```
00238 QUEUE_REFERENCE_COUNT:
                .BLKB   4
0023C MBX_MESSAGE_COUNT:
                .BLKB   4
00240 MBX:      .BLKB   4
00244 MBX_END:  .BLKB   4
00248 MEMORY_FREE_QUEUES:
                .BLKB   40
00270 NONAST_WORK_QUEUE:
                .BLKB   8
00278 BCB_FREE_LIST:
                .BLKB   4
0027C BCB_ACTIVE_LIST:
                .BLKB   4
00280 GQL_FREE_LIST:
                .BLKB   4
00284 GQL_ACTIVE_LIST:
                .BLKB   4
00288 OPEN_GETQUI_LIST:
                .BLKB   4
0028C PROCESS_DATA_LIST:
                .BLKB   4
00290 SYMBIONT_CONTROL:
                .BLKB   4
00294 SPARE_AREA:
                .BLKB   12
002A0 REMOTE_REQUEST_LKSB:
                .BLKB   8
002A8 QUEUE_FILE_LKSB:
                .BLKB   8
002B0 QUEUE_LOCK_LKSB:
                .BLKB   8
002B8 RSP:      .BLKB   8
002C0 JBC_PRIORITY:
                .BLKB   4
002C4 JBC_PRIVILEGES:
                .BLKB   8
002CC JBC_QUOTAS:
                .BLKB   66
0030E           .BLKB   2
00310 JBC_UIC:  .BLKB   4
00314 QUEUE_FAB:
                .BLKB   80
00364 QUEUE_RAB:
                .BLKB   68
003A8 QUEUE_NAM:
                .BLKB   96
00408 QUEUE_XAB:
                .BLKB   88
00460 QUEUE_RSA:
                .BLKB   255
0055F           .BLKB   1
00560 QUEUE_ALQ:
                .BLKB   4
00564 QUEUE_MBF:
                .BLKB   1
00565           .BLKB   3
```

```
00568 ACCOUNTING_FABS:
                .BLKB   8
00570 ACCOUNTING_RABS:
                .BLKB   8
00578 ACCOUNT_FAB_A:
                .BLKB   80
005C8 ACCOUNT_RAB_A:
                .BLKB   68
0060C ACCOUNT_NAM_A:
                .BLKB   96
0066C ACCOUNT_RSA_A:
                .BLKB   255
0076B           .BLKB   1
0076C ACCOUNT_FAB_B:
                .BLKB   80
007BC ACCOUNT_RAB_B:
                .BLKB   68
00800 ACCOUNT_NAM_B:
                .BLKB   96
00860 ACCOUNT_RSA_B:
                .BLKB   255
0095F           .BLKB   1
00960 DIAG_FAB:
                .BLKB   80
009B0 DIAG_RAB:
                .BLKB   68
009F4 MBX_CHAN:
                .BLKB   4
009F8 MBX_IOSB:
                .BLKB   8
00A00 MBX_BUFFER:
                .BLKB   1024
00E00 VALUE_STORAGE_BASE:
                .BLKB   0
00E00 ITEM_PRESENT:
                .BLKB   32
00E20 VALUE_GETQUI_BASE:
                .BLKB   0
00E20 VALUE_ACCOUNTING_MESSAGE:
                .BLKB   6
00E26 VALUE_ACCOUNTING_TYPES:
                .BLKB   4
00E2A VALUE_AFTER_TIME:
                .BLKB   8
00E32 VALUE_ALIGNMENT_PAGES:
                .BLKB   1
00E33 VALUE_BASE_PRIORITY:
                .BLKB   1
00E34 VALUE_BATCH_INPUT:
                .BLKB   6
00E3A VALUE_BATCH_OUTPUT:
                .BLKB   10
00E44 VALUE_BUFFER_COUNT:
                .BLKB   1
00E45 VALUE_CHARACTERISTIC_NAME:
                .BLKB   6
00E4B VALUE_CHARACTERISTIC_NUMBER:
```

```
                        .BLKB   1
00E4C VALUE_CHARACTERISTICS:
                        .BLKB   16
00E5C VALUE_CHECKPOINT_DATA:
                        .BLKB   6
00E62 VALUE_CLI:
                        .BLKB   6
00E68 VALUE_CPU_DEFAULT:
                        .BLKB   4
00E6C VALUE_CPU_LIMIT:
                        .BLKB   4
00E70 VALUE_DESTINATION_QUEUE:
                        .BLKB   8
00E78 VALUE_DEVICE_NAME:
                        .BLKB   6
00E7E VALUE_ENTRY_NUMBER:
                        .BLKB   4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
                        .BLKB   10
00E8C VALUE_EXTEND_QUANTITY:
                        .BLKB   2
00E8E VALUE_FILE_COPIES:
                        .BLKB   1
00E8F VALUE_FILE_IDENTIFICATION:
                        .BLKB   36
00EB3 VALUE_FILE_SETUP_MODULES:
                        .BLKB   6
00EB9 VALUE_FILE_SPECIFICATION:
                        .BLKB   6
00EBF VALUE_FIRST_PAGE:
                        .BLKB   4
00EC3 VALUE_FORM_DESCRIPTION:
                        .BLKB   6
00EC9 VALUE_FORM_LENGTH:
                        .BLKB   1
00ECA VALUE_FORM_MARGIN_BOTTOM:
                        .BLKB   1
00ECB VALUE_FORM_MARGIN_LEFT:
                        .BLKB   2
00ECD VALUE_FORM_MARGIN_RIGHT:
                        .BLKB   2
00ECF VALUE_FORM_MARGIN_TOP:
                        .BLKB   1
00ED0 VALUE_FORM_NAME:
                        .BLKB   6
00ED6 VALUE_FORM_NUMBER:
                        .BLKB   4
00EDA VALUE_FORM:
                        .BLKB   8
00EE2 VALUE_FORM_SETUP_MODULES:
                        .BLKB   6
00EE8 VALUE_FORM_STOCK:
                        .BLKB   6
00EEE VALUE_FORM_WIDTH:
                        .BLKB   2
00EF0 VALUE_GENERIC_TARGET:
                        .BLKB   996
```

```
012D4 VALUE_JOB_COPIES:
                .BLKB   1
012D5 VALUE_JOB_LIMIT:
                .BLKB   1
012D6 VALUE_JOB_NAME:
                .BLKB   6
012DC VALUE_JOB_RESET_MODULES:
                .BLKB   6
012E2 VALUE_JOB_SIZE_MAXIMUM:
                .BLKB   4
012E6 VALUE_JOB_SIZE_MINIMUM:
                .BLKB   4
012EA VALUE_JOB_STATUS_OUTPUT:
                .BLKB   10
012F4 VALUE_LAST_PAGE:
                .BLKB   4
012F8 VALUE_LIBRARY_SPECIFICATION:
                .BLKB   6
012FE VALUE_LOG_QUEUE:
                .BLKB   8
01306 VALUE_LOG_SPECIFICATION:
                .BLKB   6
0130C VALUE_NOTE:
                .BLKB   6
01312 VALUE_OPERATOR_REQUEST:
                .BLKB   6
01318 VALUE_OWNER_UIC:
                .BLKB   4
0131C VALUE_PAGE_SETUP_MODULES:
                .BLKB   6
01322 VALUE_PARAMETER_1:
                .BLKB   6
01328 VALUE_PARAMETER_2:
                .BLKB   6
0132E VALUE_PARAMETER_3:
                .BLKB   6
01334 VALUE_PARAMETER_4:
                .BLKB   6
0133A VALUE_PARAMETER_5:
                .BLKB   6
01340 VALUE_PARAMETER_6:
                .BLKB   6
01346 VALUE_PARAMETER_7:
                .BLKB   6
0134C VALUE_PARAMETER_8:
                .BLKB   6
01352 VALUE_PRIORITY:
                .BLKB   1
01353 VALUE_PROCESSOR:
                .BLKB   6
01359 VALUE_PROTECTION:
                .BLKB   4
0135D VALUE_QUEUE:
                .BLKB   6
01363 VALUE_QUEUE_FILE_SPECIFICATION:
                .BLKB   6
01369 VALUE_RELATIVE_PAGE:
```

```
                        .BLKB   4
0136D VALUE_RESERVED_INPUT_1:
                        .BLKB   1
0136E VALUE_RESERVED_INPUT_2:
                        .BLKB   2
01370 VALUE_RESERVED_INPUT_3:
                        .BLKB   4
01374 VALUE_RESERVED_INPUT_4:
                        .BLKB   6
0137A VALUE_RESERVED_OUTPUT_1:
                        .BLKB   10
01384 VALUE_RESERVED_OUTPUT_2:
                        .BLKB   10
0138E VALUE_SEARCH_STRING:
                        .BLKB   6
01394 VALUE_SCSNODE_NAME:
                        .BLKB   6
0139A VALUE_WSDEFAULT:
                        .BLKB   2
0139C VALUE_WSEXTENT:
                        .BLKB   2
0139E VALUE_WSQUOTA:
                        .BLKB   2
013A0 VALUE_STORAGE_END:
                        .BLKB   0

                JBC$_CLOSEOUT=      266328
                JBC$_NOCMKRNL=      272388
                JBC$_NOOPER=        272532
                JBC$_NOSYSNAM=      272404
                JBC$_OPENIN=        266392
                JBC$_OPENOUT=       266400
                JBC$_READERR=       266416
                JBC$_WRITEERR=      266448
                        .EXTRN  AFTER_AST, ALLOCATE_MEMORY
                        .EXTRN  ALLOCATE_RECORD
                        .EXTRN  BROADCAST_MESSAGE
                        .EXTRN  DEALLOCATE_RECORD_LIST
                        .EXTRN  DELETE_SJH_RECORD
                        .EXTRN  ENQUEUE_JOB, READ_RECORD
                        .EXTRN  RELEASE_RECORD, REWRITE_RECORD
                        .EXTRN  SCAN_INCOMPLETE_SERVICES
                        .EXTRN  UPDATE_GETQUI_DATA
                        .EXTRN  WRITE_ACCOUNTING_RECORD

                        .PSECT  CODE,NOWRT,2

                    000C 00000          .ENTRY  ENTER_PROCESS_DATA, Save R2,R3          ; 1159
    53 00000000'    EF   9E 00002       MOVAB   PROCESS_DATA_LIST, R3                   ;
    50              63   D0 00009       MOVL    PROCESS_DATA_LIST, PDB                  ; 1199
                    17   13 0000C 1$:    BEQL    3$                                      ; 1200
    1F        04    A0   D1 0000E       CMPL    4(PDB), #31                             ; 1202
                    0C   1E 00012       BGEQU   2$                                      ;
51        04    A0        04   78 00014       ASHL    #4, 4(PDB), R1                  ; 1205
    52        08 A140   9E 00019       MOVAB   8(R1)[PDB], PDE                         ;
                    05   11 0001E       BRB     3$                                      ; 1204
    50              60   D0 00020 2$:    MOVL    (PDB), PDB                              ; 1208
```

```
                                    E7  11 00023          BRB     1$                                      ; 1200
                                    50  D5 00025 3$:      TSTL    PDB                                     ; 1214
                                    11  12 00027          BNEQ    4$                                      ;
                         00000000G  EF      00 FB 00029          CALLS   #0, ALLOCATE_MEMORY             ; 1217
                                    60      63 D0 00030          MOVL    PROCESS_DATA_LIST, (PDB)        ; 1218
                                    63      50 D0 00033          MOVL    PDB, PROCESS_DATA_LIST          ; 1219
                                    52  08  A0 9E 00036          MOVAB   8(R0), PDE                      ; 1220
                                        04  A0 D6 0003A 4$:      INCL    4(PDB)                          ; 1226
                                04  A2  04  AC D0 0003D          MOVL    TYPE, 4(PDE)                    ; 1227
                                    62  08  AC D0 00042          MOVL    PID, (PDE)                      ; 1228
                                    03      6C 91 00046          CMPB    (AP), #3                        ; 1229
                                            05 1F 00049          BLSSU   5$                              ;
                                08  A2  0C  AC D0 0004B          MOVL    P1, 8(PDE)                      ;
                                    04      6C 91 00050 5$:      CMPB    (AP), #4                        ; 1230
                                            05 1F 00053          BLSSU   6$                              ;
                                0C  A2  10  AC D0 00055          MOVL    P2, 12(PDE)                     ;
                                            04 0005A 6$:         RET                                     ; 1231

; Routine Size:  91 bytes,    Routine Base:  CODE + 0000
```

```
;  195          1232  1 GLOBAL ROUTINE FIND_PROCESS_DATA(TYPE,PID,REMOVE; TY,P1,P2): L_OUTPUT_3=
;  196          1233  1
;  197          1234  1 !++
;  198          1235  1 !
;  199          1236  1 ! FUNCTIONAL DESCRIPTION:
;  200          1237  1 !       This routine looks up an entry in the process data structure.
;  201          1238  1 !
;  202          1239  1 ! INPUT PARAMETERS:
;  203          1240  1 !       TYPE            - Type of process.
;  204          1241  1 !       PID             - Process ID.
;  205          1242  1 !       REMOVE          - True if entry to be removed.
;  206          1243  1 !
;  207          1244  1 ! IMPLICIT INPUTS:
;  208          1245  1 !       NONE
;  209          1246  1 !
;  210          1247  1 ! OUTPUT PARAMETERS:
;  211          1248  1 !       TY              - Type of process found.
;  212          1249  1 !       P1              - First parameter.
;  213          1250  1 !       P2              - Second parameter.
;  214          1251  1 !
;  215          1252  1 ! IMPLICIT OUTPUTS:
;  216          1253  1 !       NONE
;  217          1254  1 !
;  218          1255  1 ! ROUTINE VALUE:
;  219          1256  1 !       True if the entry was found, false otherwise.
;  220          1257  1 !
;  221          1258  1 ! SIDE EFFECTS:
;  222          1259  1 !       NONE
;  223          1260  1 !
;  224          1261  1 !--
;  225          1262  1
;  226          1263  2 BEGIN
;  227          1264  2 LOCAL
;  228          1265  2         PDB:            REF BBLOCK;     ! Pointer to PDB
;  229          1266  2
;  230          1267  2
;  231          1268  2 PDB = .PROCESS_DATA_LIST;
;  232          1269  2 WHILE .PDB NEQ 0 DO
;  233          1270  3     BEGIN
;  234          1271  3     LOCAL
;  235          1272  3         PDE:            REF BBLOCK;     ! Pointer to PDB entry
;  236          1273  3
;  237          1274  3     PDE = PDB[PDB_ENTRIES];
;  238          1275  3     INCR CBN FROM 0 TO .PDB[PDB_COUNT]-1 DO
;  239          1276  4         BEGIN
;  240          1277  4         IF .PDE[PDE_PID] EQL .PID
;  241          1278  5         AND (.TYPE EQL PDE_K_ANY OR .TYPE EQL .PDE[PDE_TYPE])
;  242          1279  4         THEN
;  243          1280  5             BEGIN
;  244          1281  5             TY = .PDE[PDE_TYPE];
;  245          1282  5             P1 = .PDE[PDE_P1];
;  246          1283  5             P2 = .PDE[PDE_P2];
;  247          1284  5             IF .REMOVE
;  248          1285  5             THEN
;  249          1286  6                 BEGIN
;  250          1287  6                 PDB[PDB_COUNT] = .PDB[PDB_COUNT] - 1;
;  251          1288  6                 CH$COPY(
```

```
;  252     1289  6                    (.PDB[PDB_COUNT] - .CBN) * PDE_S_ENTRY,
;  253     1290  6                    .PDE + PDE_S_ENTRY,
;  254     1291  6                    0,
;  255     1292  6                    (.PDB[PDB_COUNT] - .CBN) * PDE_S_ENTRY + PDE_S_ENTRY,
;  256     1293  6                    .PDE);
;  257     1294  5                END;
;  258     1295  5            RETURN TRUE;
;  259     1296  4            END;
;  260     1297  4        PDE = .PDE + PDE_S_ENTRY;
;  261     1298  3        END;
;  262     1299  3    PDB = .PDB[PDB_LINK];
;  263     1300  2    END;
;  264     1301  2
;  265     1302  2
;  266     1303  2 FALSE
;  267     1304  1 END;


                                    01FC 00000          .ENTRY  FIND_PROCESS_DATA, Save R2,R3,R4,R5,R6,R7,- ; 1232
                                                                R8                                          ;
                                5E          04  C2 00002        SUBL2   #4, SP                              ;
                                57 00000000'    EF  D0 00005        MOVL    PROCESS_DATA_LIST, PDB          ; 1268
                                                51  13 0000C 1$:    BEQL    7$                              ; 1269
                                56          08  A7  9E 0000E        MOVAB   8(R7), PDE                      ; 1274
                                6E          04  A7  D0 00012        MOVL    4(PDB), (SP)                    ; 1275
                                58              01  CE 00016        MNEGL   #1, CBN                         ;
                                                3B  11 00019        BRB     6$                              ;
                                        08      AC      66  D1 0001B 2$:    CMPL    (PDE), PID              ; 1277
                                                32  12 0001F        BNEQ    5$                              ;
                                            04  AC      D5  00021        TSTL    TYPE                        ; 1278
                                                07  13 00024        BEQL    3$                              ;
                                        04      A6  04  AC  D1 00026        CMPL    TYPE, 4(PDE)            ;
                                                26  12 0002B        BNEQ    5$                              ;
                                    59          04  A6  7D 0002D 3$:    MOVQ    4(PDE), TY                  ; 1281
                                    5B          0C  A6  D0 00031        MOVL    12(PDE), P2                 ; 1283
                                    16          0C  AC  E9 00035        BLBC    REMOVE, 4$                  ; 1284
                                            04  A7  D7 00039        DECL    4(PDB)                          ; 1287
                        50          04      A7      58  C3 0003C        SUBL3   CBN, 4(PDB), R0             ; 1289
                                    50              10  C4 00041        MULL2   #16, R0                     ;
                                    51          10  A0  9E 00044        MOVAB   16(R0), R1                  ; 1292
                51      00          10      A6      50  2C 00048        MOVC5   R0, 16(PDE), #0, R1, (PDE)  ; 1293
                                                    66      0004E                                           ;
                                    50              01  D0 0004F 4$:    MOVL    #1, R0                      ; 1295
                                                04  00052        RET                                     ;
                                    56              10  C0 00053 5$:    ADDL2   #16, PDE                    ; 1297
                        C1          58              6E  F2 00056 6$:    AOBLSS  (SP), CBN, 2$               ; 1275
                                    57              67  D0 0005A        MOVL    (PDB), PDB                  ; 1299
                                                    AD  11 0005D        BRB     1$                          ; 1269
                                                    50  D4 0005F 7$:    CLRL    R0                          ; 1304
                                                04  00061        RET                                     ;

; Routine Size:  98 bytes,    Routine Base:  CODE + 005B
```

```
;  269     1305  1 GLOBAL ROUTINE SEARCH_QUEUES(
;  270     1306  1         QSM,
;  271     1307  1         SMQ_NF, SMQ_F,
;  272     1308  1         ENTRY, JOBNAME,
;  273     1309  1         ACCESS_CHECK,
;  274     1310  1         REMOVE,
;  275     1311  1         CTX;
;  276     1312  1         SJH_N, SJH, SMQ_N, SMQ): L_OUTPUT_4=
;  277     1313  1
;  278     1314  1 !++
;  279     1315  1 !
;  280     1316  1 ! FUNCTIONAL DESCRIPTION:
;  281     1317  1 !       This routine provides a general facility to search the job queues.
;  282     1318  1 !
;  283     1319  1 ! INPUT PARAMETERS:
;  284     1320  1 !
;  285     1321  1 !       QSM             - Bit mask that identifies queues to be searched.
;  286     1322  1 !
;  287     1323  1 !       SMQ_NF          - Record number of SMQ to search.
;  288     1324  1 !       SMQ_F           - Pointer to SMQ to search.
;  289     1325  1 !
;  290     1326  1 !       ENTRY           - Address of job entry number, or 0 to denote wild.
;  291     1327  1 !
;  292     1328  1 !       JOBNAME         - Short descriptor for job name, or 0 to denote wild.
;  293     1329  1 !                         This parameter is significant only if ENTRY is 0.
;  294     1330  1 !                         Job name is implicitly qualified by username.
;  295     1331  1 !
;  296     1332  1 !       ACCESS_CHECK    - Address of access check routine, or 0 to denote none.
;  297     1333  1 !
;  298     1334  1 !       REMOVE          - Specifies if job to be removed from queue.
;  299     1335  1 !                               QSM_K_NO_REMOVE          Never remove
;  300     1336  1 !                               QSM_K_REMOVE             Always remove
;  301     1337  1 !                               QSM_K_REMOVE_INACTIVE    Remove unless executing
;  302     1338  1 !
;  303     1339  1 !       CTX             - Pointer to context area of size QSM_K_CTXSIZE bytes
;  304     1340  1 !                         for wildcard operations (optional).  Initialize to
;  305     1341  1 !                         binary zeros prior to first call.
;  306     1342  1 !
;  307     1343  1 ! IMPLICIT INPUTS:
;  308     1344  1 !       MBX             - Pointer to buffered mailbox message.
;  309     1345  1 !
;  310     1346  1 ! OUTPUT PARAMETERS:
;  311     1347  1 !       SJH_N           - Record number of SJH.
;  312     1348  1 !       SJH             - Pointer to SJH.
;  313     1349  1 !       SMQ_N           - Record number of SQH or SMQ.
;  314     1350  1 !       SMQ             - Pointer to SQH or SMQ.
;  315     1351  1 !
;  316     1352  1 ! IMPLICIT OUTPUTS:
;  317     1353  1 !       NONE
;  318     1354  1 !
;  319     1355  1 ! ROUTINE VALUE:
;  320     1356  1 !       SS$_NORMAL      - Job found.
;  321     1357  1 !       JBC$_NOSUCHJOB  - Job not found.
;  322     1358  1 !       JBC$_NOPRIV     - No privilege to operate on job.
;  323     1359  1 !
;  324     1360  1 ! SIDE EFFECTS:
;  325     1361  1 !       NONE
```

```
;  326     1362  1 !
;  327     1363  1 !--
;  328     1364  1
;  329     1365  2 BEGIN
;  330     1366  2 MAP
;  331     1367  2         QSM:            BBLOCK,                 ! Queue search bitmask
;  332     1368  2         SMQ_F:          REF BBLOCK,             ! Pointer to SMQ
;  333     1369  2         ENTRY:          REF VECTOR[,WORD],      ! Pointer to job ID or 0
;  334     1370  2         JOBNAME:        REF BBLOCK,             ! Descriptor for name or 0
;  335     1371  2         CTX:            REF VECTOR,             ! Pointer to context block
;  336     1372  2         SJH:            REF BBLOCK,             ! Pointer to SJH
;  337     1373  2         SMQ:            REF BBLOCK;             ! Pointer to SQH or SMQ
;  338     1374  2 LOCAL
;  339     1375  2         LIST_OFFSET,                            ! Offset to list head in SQH or SMQ
;  340     1376  2         QID,                                    ! Queue type context
;  341     1377  2         SQX_N,                                  ! Record number of SQX
;  342     1378  2         SQX:            REF BBLOCK,             ! Pointer to SQX
;  343     1379  2         SQE_N,                                  ! Offset to SQX entry
;  344     1380  2         SJH_NP,                                 ! Record number of predecessor of SJH
;  345     1381  2         SJH_P:          REF BBLOCK;             ! Pointer to predecessor of SJH
;  346     1382  2 BUILTIN
;  347     1383  2         NULLPARAMETER;
;  348     1384  2
;  349     1385  2
;  350     1386  2 ! Set up context for the search.  If the context block is supplied, initialize
;  351     1387  2 ! context from the block; otherwise, initialize as for first call.
;  352     1388  2 !
;  353     1389  2 LIST_OFFSET = 0;
;  354     1390  2 QID = 0;
;  355     1391  2 SQX_N = 0;
;  356     1392  2 SQX = 0;
;  357     1393  2 SQE_N = 0;
;  358     1394  2 SMQ_N = 0;
;  359     1395  2 SMQ = 0;
;  360     1396  2 SJH_NP = 0;
;  361     1397  2 SJH_P = 0;
;  362     1398  2 IF NOT NULLPARAMETER(8)
;  363     1399  2 THEN
;  364     1400  3     BEGIN
;  365     1401  3     LIST_OFFSET =       .CTX[0];
;  366     1402  3     QID =               .CTX[1];
;  367     1403  3     SQX_N =             .CTX[2];
;  368     1404  3     SQX =               .CTX[3];
;  369     1405  3     SQE_N =             .CTX[4];
;  370     1406  3     SMQ_N =             .CTX[5];
;  371     1407  3     SMQ =               .CTX[6];
;  372     1408  3     SJH_NP =            .CTX[7];
;  373     1409  3     SJH_P =             .CTX[8];
;  374     1410  2     END;
;  375     1411  2
;  376     1412  2
;  377     1413  2 ! Loop until a job is found, or until all queues have been searched.
;  378     1414  2 !
;  379     1415  2 WHILE TRUE DO
;  380     1416  3     BEGIN
;  381     1417  3
;  382     1418  3     ! If a new queue needs to be started, find the next queue that must be
```

```
;  383     1419  3      ! searched.  If no more queues, return failure.
;  384     1420  3      !
;  385     1421  3      IF .LIST_OFFSET EQL 0
;  386     1422  3      THEN
;  387     1423  4          BEGIN
;  388     1424  4
;  389     1425  4          ! Loop that advances over queues until one that is selected by the
;  390     1426  4          ! queue selection criteria (QSM and SMQ) is found.
;  391     1427  4          !
;  392     1428  4          WHILE TRUE DO
;  393     1429  5              BEGIN
;  394     1430  5
;  395     1431  5              ! Advance to next queue type.
;  396     1432  5              !
;  397     1433  5              QID = .QID + 1;
;  398     1434  5
;  399     1435  5
;  400     1436  5              ! Case on the QID context to select the next queue type.
;  401     1437  5              !
;  402     1438  5              CASE .QID FROM 1 TO 3 OF
;  403     1439  5                  SET
;  404     1440  5
;  405     1441  5
;  406     1442  5                  [1]:    ! open queue
;  407     1443  6                      BEGIN
;  408     1444  6                      IF .QSM[QSM_V_OPEN]
;  409     1445  6                      AND .SMQ_F[SMQ$W_OPEN_JOB_COUNT] NEQ 0
;  410     1446  6                      THEN
;  411     1447  7                          BEGIN
;  412     1448  7                          LIST_OFFSET = $BYTEOFFSET(SQH$L_OPEN_LIST);
;  413     1449  7                          EXITLOOP;
;  414     1450  6                          END;
;  415     1451  5                      END;
;  416     1452  5
;  417     1453  5
;  418     1454  5                  [2]:    ! timer queue
;  419     1455  6                      BEGIN
;  420     1456  6                      IF .QSM[QSM_V_TIMER]
;  421     1457  6                      AND .SMQ_F[SMQ$W_TIMER_JOB_COUNT] NEQ 0
;  422     1458  6                      THEN
;  423     1459  7                          BEGIN
;  424     1460  7                          LIST_OFFSET = $BYTEOFFSET(SQH$L_TIMER_LIST);
;  425     1461  7                          EXITLOOP;
;  426     1462  6                          END;
;  427     1463  5                      END;
;  428     1464  5
;  429     1465  5
;  430     1466  5                  [3]:    ! pending queue
;  431     1467  6                      BEGIN
;  432     1468  6                      IF .QSM[QSM_V_PENDING]
;  433     1469  6                      AND .SMQ_F[SMQ$W_PENDING_JOB_COUNT] NEQ 0
;  434     1470  6                      THEN
;  435     1471  7                          BEGIN
;  436     1472  7                          IF .SMQ_F[SMQ$V_BATCH]
;  437     1473  7                              THEN LIST_OFFSET = $BYTEOFFSET(SQH$L_PENDING_BATCH_LIST)
;  438     1474  7                              ELSE LIST_OFFSET = $BYTEOFFSET(SQH$L_PENDING_PRINT_LIST);
;  439     1475  7                          EXITLOOP;
```

```
;  440          1476  6                        END;
;  441          1477  5                    END;
;  442          1478  5
;  443          1479  5
;  444          1480  5                [OUTRANGE]:
;  445          1481  6                    BEGIN
;  446          1482  6                    IF NOT .QID
;  447          1483  6                    THEN
;  448          1484  7                        BEGIN
;  449          1485  7                        LOCAL
;  450          1486  7                            SMQ_NP;     ! Predecessor of new SMQ_N
;  451          1487  7                        LABEL
;  452          1488  7                            FIND_SELECTED;
;  453          1489  7
;  454          1490  7
;  455          1491  7                        ! Even value greater than 3; establish a new queue
;  456          1492  7                        ! header.
;  457          1493  7                        !
;  458          1494  7                        IF .QSM[QSM_V_CURRENT]
;  459          1495  7                        THEN
;  460          1496  8 FIND_SELECTED:              BEGIN
;  461          1497  8
;  462          1498  8                            ! Read the queue header if not yet done.
;  463          1499  8                            !
;  464          1500  8                            IF .SMQ_N EQL 0
;  465          1501  8                            THEN
;  466          1502  8                                SMQ = READ_RECORD(SMQ_N = SQH$K_RECNO);
;  467          1503  8
;  468          1504  8
;  469          1505  8                            ! Including current queues; the requested queue and
;  470          1506  8                            ! all execution queues of the same type must be
;  471          1507  8                            ! examined.  Execute a scan of the queue index to
;  472          1508  8                            ! locate these queues.
;  473          1509  8                            !
;  474          1510  8                            SMQ_NP = .SMQ_N;
;  475          1511  8                            SMQ_N = 0;
;  476          1512  8
;  477          1513  8
;  478          1514  8                            ! Initialize to the first queue index block.
;  479          1515  8                            !
;  480          1516  8                            IF .SQX_N EQL 0
;  481          1517  8                            THEN
;  482          1518  9                                BEGIN
;  483          1519  9                                SQX_N = .SMQ[SQH$L_QUEUE_INDEX_LIST];
;  484          1520  9                                SQE_N = $BYTEOFFSET(SYM$T_DATA) - SQX$S_SQX;
;  485          1521  8                                END;
;  486          1522  8
;  487          1523  8
;  488          1524  8                            ! Loop over queue index blocks.
;  489          1525  8                            !
;  490          1526  8                            WHILE .SQX_N NEQ 0 DO
;  491          1527  9                                BEGIN
;  492          1528  9                                LOCAL
;  493          1529  9                                    SQX_NS;
;  494          1530  9
;  495          1531  9
;  496          1532  9                                ! Read the record if this has not been done.
```

```
;  497          1533  9                    !
;  498          1534  9                    IF .SQX EQL 0 THEN SQX = READ_RECORD(.SQX_N);
;  499          1535  9
;  500          1536  9
;  501          1537  9                    ! Loop over queue index entries.
;  502          1538  9                    !
;  503          1539  9                    WHILE TRUE DO
;  504          1540 10                        BEGIN
;  505          1541 10                        LOCAL
;  506          1542 10                            SQE:            REF BBLOCK;
;  507          1543 10
;  508          1544 10
;  509          1545 10                        ! Advance to next entry, and ensure that it
;  510          1546 10                        ! is valid.
;  511          1547 10                        !
;  512          1548 10                        SQE_N = .SQE_N + SQX$S_SQX;
;  513          1549 10                        IF .SQE_N GEQU $BYTEOFFSET(SYM$T_DATA) + SQX$S_SQX * SQX$K_ENTRIES
;  514          1550 10                            THEN EXITLOOP;
;  515          1551 10                        SQE = .SQX + .SQE_N;
;  516          1552 10                        IF CH$RCHAR(SQE[SQX$T_NAME]) EQL 0 THEN EXITLOOP;
;  517          1553 10
;  518          1554 10
;  519          1555 10                        ! Determine if this queue is interesting.
;  520          1556 10                        !
;  521          1557 10                        IF .SQE[SQX$L_QUEUE_LINK] EQL .SMQ_NF
;  522          1558 11                        OR (.SQE[SQX$V_BATCH] EQL .SMQ_F[SMQ$V_BATCH]
;  523          1559 11                            AND .SQE[SQX$V_EXECUTOR])
;  524          1560 10                        THEN
;  525          1561 11                            BEGIN
;  526          1562 11                            SMQ_N = .SQE[SQX$L_QUEUE_LINK];
;  527          1563 11                            LEAVE FIND_SELECTED;
;  528          1564 10                            END;
;  529          1565  9                        END;
;  530          1566  9
;  531          1567  9
;  532          1568  9                        ! Advance to next index record.
;  533          1569  9                        !
;  534          1570  9                        SQX_NS = .SQX[SYM$L_LINK];
;  535          1571  9                        RELEASE_RECORD(.SQX_N);
;  536          1572  9                        SQX = 0;
;  537          1573  9                        SQX_N = .SQX_NS;
;  538          1574  9                        SQE_N = $BYTEOFFSET(SYM$T_DATA) - SQX$S_SQX;
;  539          1575  8                        end;
;  540          1576  8                    END
;  541          1577  7                ELSE
;  542          1578  8                    BEGIN
;  543          1579  8
;  544          1580  8                    ! Excluding current queues; only the requested queue
;  545          1581  8                    ! must be examined.
;  546          1582  8                    !
;  547          1583  8                    SMQ_NP = .SMQ_N;
;  548          1584  8                    IF .SMQ_N LEQU SQH$K_RECNO
;  549          1585  8                        THEN SMQ_N = .SMQ_NF
;  550          1586  8                        ELSE SMQ_N = 0;
;  551          1587  7                    END;
;  552          1588  7
;  553          1589  7
```

```
;  554     1590  7                    ! Release the previous queue header, and read the next.
;  555     1591  7                    ! If no more queues, return with failure.
;  556     1592  7                    !
;  557     1593  7                    IF .SMQ_NP NEQ 0 THEN RELEASE_RECORD(.SMQ_NP);
;  558     1594  7                    IF .SMQ_N EQL 0 THEN RETURN JBC$_NOSUCHJOB;
;  559     1595  7                    SMQ = READ_RECORD(.SMQ_N);
;  560     1596  7
;  561     1597  7
;  562     1598  7                    ! Now process hold job queue of the queue header just
;  563     1599  7                    ! established.
;  564     1600  7                    !
;  565     1601  7                    IF .QSM[QSM_V_HOLD]
;  566     1602  7                    AND .SMQ_NF EQL .SMQ_N
;  567     1603  7                    AND .SMQ[SMQ$L_HOLD_LIST] NEQ 0
;  568     1604  7                    THEN
;  569     1605  8                        BEGIN
;  570     1606  8                        LIST_OFFSET = $BYTEOFFSET(SMQ$L_HOLD_LIST);
;  571     1607  8                        EXITLOOP;
;  572     1608  7                        END;
;  573     1609  7                END
;  574     1610  6            ELSE
;  575     1611  7                BEGIN
;  576     1612  7
;  577     1613  7                ! Odd value greater than 3; current job queue of the
;  578     1614  7                ! queue header established by the previous value.
;  579     1615  7                !
;  580     1616  7                IF .QSM[QSM_V_CURRENT]
;  581     1617  7                AND .SMQ_F[SMQ$V_BATCH] EQL .SMQ[SMQ$V_BATCH]
;  582     1618  7                AND .SMQ[SMQ$L_CURRENT_LIST] NEQ 0
;  583     1619  7                THEN
;  584     1620  8                    BEGIN
;  585     1621  8                    LIST_OFFSET = $BYTEOFFSET(SMQ$L_CURRENT_LIST);
;  586     1622  8                    EXITLOOP;
;  587     1623  7                    END;
;  588     1624  6                END;
;  589     1625  5                END;
;  590     1626  5
;  591     1627  5
;  592     1628  5                TES;
;  593     1629  4            END;
;  594     1630  4
;  595     1631  4
;  596     1632  4        IF .SMQ_N EQL 0 THEN SMQ = READ_RECORD(SMQ_N = SQH$K_RECNO);
;  597     1633  4        SJH_NP = .SMQ_N;
;  598     1634  4        SJH_P = 0;
;  599     1635  4        SJH_N = .SMQ[.LIST_OFFSET,0,32,0];
;  600     1636  4        END
;  601     1637  3    ELSE
;  602     1638  3        IF .SJH_P EQL 0
;  603     1639  3            THEN SJH_N = .SMQ[.LIST_OFFSET,0,32,0]
;  604     1640  3            ELSE SJH_N = .SJH_P[SYM$L_LINK];
;  605     1641  3
;  606     1642  3
;  607     1643  3    ! Now search the queue.
;  608     1644  3    !
;  609     1645  3    WHILE .SJH_N NEQ 0 DO
;  610     1646  4        BEGIN
```

```
;  611     1647  4          SJH = READ_RECORD(.SJH_N);
;  612     1648  4          IF
;  613     1649  5              BEGIN
;  614     1650  5              .SMQ_NF EQL .SJH[SJH$L_QUEUE_LINK] OR
;  615     1651  6              (.QID GTRU 3 AND .QID AND .SMQ_NF EQL .SMQ_N)
;  616     1652  5              END
;  617     1653  4          AND
;  618     1654  5              BEGIN
;  619     1655  5              IF .ENTRY NEQ 0
;  620     1656  5              THEN
;  621     1657  5                  .ENTRY[0] EQL .SJH[SYM$L_ENTRY_NUMBER]
;  622     1658  5              ELSE
;  623     1659  5                  IF .JOBNAME EQL 0
;  624     1660  5                  THEN
;  625     1661  5                      TRUE
;  626     1662  5                  ELSE
;  627     1663  5                      IF .JOBNAME[SDSC_W_LENGTH] EQL CH$RCHAR(SJH[SJH$T_NAME])
;  628     1664  5                      THEN
;  629     1665  5                          CH$EQL(
;  630     1666  5                              .JOBNAME[SDSC_W_LENGTH], .JOBNAME[SDSC_A_POINTER],
;  631     1667  5                              .JOBNAME[SDSC_W_LENGTH], SJH[SJH$T_NAME]+1) AND
;  632     1668  5                          CH$EQL(
;  633     1669  5                              ACM$S_USERNAME, MBX[ACM$T_USERNAME],
;  634     1670  5                              SJH$S_USERNAME, SJH[SJH$T_USERNAME])
;  635     1671  5                      ELSE
;  636     1672  5                          FALSE
;  637     1673  5              END
;  638     1674  4          THEN
;  639     1675  5              BEGIN
;  640     1676  5              LOCAL
;  641     1677  5                  REMOVING;                   ! True if removing this entry
;  642     1678  5
;  643     1679  5
;  644     1680  5              ! If an access check was requested, execute it.
;  645     1681  5              !
;  646     1682  5              IF .ACCESS_CHECK NEQ 0
;  647     1683  5              THEN
;  648     1684  5                  IF NOT (.ACCESS_CHECK)(.SMQ_F, .SJH)
;  649     1685  5                  THEN
;  650     1686  6                      BEGIN
;  651     1687  6                      IF .SQX_N NEQ 0
;  652     1688  6                          THEN RELEASE_RECORD(.SQX_N);
;  653     1689  6                      IF .SMQ_N NEQ 0 AND .SMQ_N NEQ .SJH_NP
;  654     1690  6                          THEN RELEASE_RECORD(.SMQ_N);
;  655     1691  6                      IF .SJH_NP NEQ 0
;  656     1692  6                          THEN RELEASE_RECORD(.SJH_NP);
;  657     1693  6                      RELEASE_RECORD(.SJH_N);
;  658     1694  6                      RETURN JBC$_NOPRIV;
;  659     1695  5                      END;
;  660     1696  5
;  661     1697  5              ! Determine if we must remove the job, based on the input parameter
;  662     1698  5              ! and whether the job is executing.
;  663     1699  5              !
;  664     1700  5              REMOVING = TRUE;
;  665     1701  5              CASE .REMOVE FROM QSM_K_NO_REMOVE TO QSM_K_REMOVE_INACTIVE OF
;  666     1702  5                  SET
;  667     1703  5
```

```
;  668     1704  5                [QSM_K_NO_REMOVE]:
;  669     1705  5                    REMOVING = FALSE;
;  670     1706  5                [QSM_K_REMOVE]:
;  671     1707  5                    0;
;  672     1708  5                [QSM_K_REMOVE_INACTIVE]:
;  673     1709  5                    IF .SJH[SJH$V_EXECUTING] THEN REMOVING = FALSE;
;  674     1710  5                TES;
;  675     1711  5
;  676     1712  5
;  677     1713  5            IF .REMOVING
;  678     1714  5            THEN
;  679     1715  6                BEGIN
;  680     1716  6
;  681     1717  6                ! Adjust the job reference counts for queues linked from the
;  682     1718  6                ! queue header.
;  683     1719  6                !
;  684     1720  6                IF .QID LEQU 3
;  685     1721  6                THEN
;  686     1722  7                    BEGIN
;  687     1723  7                    CASE .QID FROM 1 TO 3 OF
;  688     1724  7                        SET
;  689     1725  7
;  690     1726  7                        [OUTRANGE]:
;  691     1727  7                            0;
;  692     1728  7
;  693     1729  7                        [1]:
;  694     1730  7                            SMQ_F[SMQ$W_OPEN_JOB_COUNT] =
;  695     1731  7                                .SMQ_F[SMQ$W_OPEN_JOB_COUNT] - 1;
;  696     1732  7
;  697     1733  7                        [2]:
;  698     1734  7                            SMQ_F[SMQ$W_TIMER_JOB_COUNT] =
;  699     1735  7                                .SMQ_F[SMQ$W_TIMER_JOB_COUNT] - 1;
;  700     1736  7
;  701     1737  7                        [3]:
;  702     1738  7                            SMQ_F[SMQ$W_PENDING_JOB_COUNT] =
;  703     1739  7                                .SMQ_F[SMQ$W_PENDING_JOB_COUNT] - 1;
;  704     1740  7
;  705     1741  7                        TES;
;  706     1742  7                    READ_RECORD(.SMQ_NF);
;  707     1743  7                    REWRITE_RECORD(.SMQ_NF);
;  708     1744  6                    END;
;  709     1745  6
;  710     1746  6
;  711     1747  6                ! Unlink the job.
;  712     1748  6                !
;  713     1749  6                UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  714     1750  6                IF .SJH_P EQL 0
;  715     1751  6                THEN
;  716     1752  7                    BEGIN
;  717     1753  7                    SMQ[.LIST_OFFSET,0,32,0] = .SJH[SYM$L_LINK];
;  718     1754  7                    IF .SJH[SYM$L_LINK] EQL 0
;  719     1755  7                        THEN SMQ[.LIST_OFFSET+4,0,32,0] = 0;
;  720     1756  7                    READ_RECORD(.SMQ_N);
;  721     1757  7                    REWRITE_RECORD(.SMQ_N);
;  722     1758  7                    IF .QID EQL 2
;  723     1759  7                    THEN
;  724     1760  8                        BEGIN
```

```
;  725     1761  8                    LOCAL
;  726     1762  8                        SJH_N2,                         ! Record number of next
;  727     1763  8                        SJH_2:      REF BBLOCK,         ! Pointer to next
;  728     1764  8                        STATUS;                         ! Status return
;  729     1765  8
;  730     1766  8                    $CANTIM(REQIDT=JBC$K_AFTER_IDT);
;  731     1767  8                    IF .SJH[SYM$L_LINK] NEQ 0
;  732     1768  8                    THEN
;  733     1769  9                        BEGIN
;  734     1770  9                        SJH_2 = READ_RECORD(SJH_N2 = .SJH[SYM$L_LINK]);
;  735   P 1771  9                        STATUS = $SETIMR(
;  736   P 1772  9                            DAYTIM=SJH_2[SJH$Q_AFTER_TIME],
;  737   P 1773  9                            ASTADR=AFTER_AST,
;  738     1774  9                            REQIDT=JBC$K_AFTER_IDT);
;  739     1775  9                        IF NOT .STATUS
;  740     1776  9                        THEN
;  741     1777  9                            SIGNAL(JBC$_SETIMR OR STS$K_ERROR, 0, .STATUS);
;  742     1778  9                        RELEASE_RECORD(.SJH_N2);
;  743     1779  8                        END;
;  744     1780  7                    END
;  745     1781  7                END
;  746     1782  6            ELSE
;  747     1783  7                BEGIN
;  748     1784  7                SJH_P[SYM$L_LINK] = .SJH[SYM$L_LINK];
;  749     1785  7                IF .SJH[SYM$L_LINK] EQL 0
;  750     1786  7                THEN
;  751     1787  8                    BEGIN
;  752     1788  8                    SMQ[.LIST_OFFSET*4,0,32,0] = .SJH_NP;
;  753     1789  8                    READ_RECORD(.SMQ_N);
;  754     1790  8                    REWRITE_RECORD(.SMQ_N);
;  755     1791  7                    END;
;  756     1792  7                READ_RECORD(.SJH_NP);
;  757     1793  7                REWRITE_RECORD(.SJH_NP);
;  758     1794  6                END;
;  759     1795  6            END
;  760     1796  5        ELSE
;  761     1797  6            BEGIN
;  762     1798  6            IF .SJH_NP NEQ .SMQ_N THEN RELEASE_RECORD(.SJH_NP);
;  763     1799  5            END;
;  764     1800  5
;  765     1801  5
;  766     1802  5        IF NOT NULLPARAMETER(8)
;  767     1803  5        THEN
;  768     1804  6            BEGIN
;  769     1805  6            CTX[0] = .LIST_OFFSET;
;  770     1806  6            CTX[1] = .QID;
;  771     1807  6            CTX[2] = .SQX_N;
;  772     1808  6            CTX[3] = .SQX;
;  773     1809  6            CTX[4] = .SQE_N;
;  774     1810  6            CTX[5] = .SMQ_N;
;  775     1811  6            CTX[6] = .SMQ;
;  776     1812  6            IF .REMOVING
;  777     1813  6            THEN
;  778     1814  7                BEGIN
;  779     1815  7                CTX[7] = .SJH_NP;
;  780     1816  7                CTX[8] = .SJH_P;
;  781     1817  7                END
```

```
;  782     1818  6                    ELSE
;  783     1819  7                        BEGIN
;  784     1820  7                        CTX[7] = .SJH_N;
;  785     1821  7                        CTX[8] = .SJH;
;  786     1822  6                        END;
;  787     1823  5                    END;
;  788     1824  5
;  789     1825  5
;  790     1826  5                RETURN SS$_NORMAL;
;  791     1827  4                END;
;  792     1828  4
;  793     1829  4
;  794     1830  4            IF .SJH_NP NEQ .SMQ_N THEN RELEASE_RECORD(.SJH_NP);
;  795     1831  4            SJH_NP = .SJH_N;
;  796     1832  4            SJH_P = .SJH;
;  797     1833  4            SJH_N = .SJH[SYM$L_LINK];
;  798     1834  3            END;
;  799     1835  3
;  800     1836  3
;  801     1837  3        ! Indicate no current queue.
;  802     1838  3        !
;  803     1839  3        IF .SJH_NP NEQ .SMQ_N THEN RELEASE_RECORD(.SJH_NP);
;  804     1840  3        LIST_OFFSET = 0;
;  805     1841  2        END;
;  806     1842  2 0
;  807     1843  1 END;


                                                .EXTRN  SYS$CANTIM, SYS$SETIMR

                        00FC 00000              .ENTRY  SEARCH_QUEUES, Save R2,R3,R4,R5,R6,R7       ; 1305
                     7E  7C 00002               CLRQ    SQX                                         ; 1392
                     7E  D4 00004               CLRL    SQE_N                                       ; 1393
                     5A  7C 00006               CLRQ    SMQ_N                                       ; 1394
                     54  D4 00008               CLRL    SMQ                                         ; 1395
                     59  D4 0000A               CLRL    SJH_NP                                      ; 1396
                     56  7C 0000C               CLRQ    LIST_OFFSET                                 ; 1389
               08    6C  91 0000E               CMPB    (AP), #8                                    ; 1398
                     2E  1F 00011               BLSSU   1$                                          ;
           20        AC  D5 00013               TSTL    32(AP)                                      ;
                     29  13 00016               BEQL    1$                                          ;
           50  20    AC  D0 00018               MOVL    CTX, R0                                     ; 1401
           56        60  D0 0001C               MOVL    (R0), LIST_OFFSET                           ;
     08    AE  04    A0  D0 0001F               MOVL    4(R0), QID                                  ; 1402
     5B    08    A0  D0 00024                   MOVL    8(R0), SQX_N                                ; 1403
     04    AE  0C    A0  D0 00028               MOVL    12(R0), SQX                                 ; 1404
     6E    10    A0  D0 0002D                   MOVL    16(R0), SQE_N                               ; 1405
     5A    14    A0  D0 00031                   MOVL    20(R0), SMQ_N                               ; 1406
     54    18    A0  D0 00035                   MOVL    24(R0), SMQ                                 ; 1407
     59    1C    A0  D0 00039                   MOVL    28(R0), SJH_NP                              ; 1408
     57    20    A0  D0 0003D                   MOVL    32(R0), SJH_P                               ; 1409
                     56  D5 00041 1$:            TSTL    LIST_OFFSET                                 ; 1421
                     03  13 00043               BEQL    2$                                          ;
                   0170  31 00045               BRW     26$                                         ;
                     08    AE  D6 00048 2$:        INCL    QID                                         ; 1433
     02        01    08    AE  CF 0004B           CASEL   QID, #1, #2                                 ; 1438
```

```
                012D            0118            0104     00050 3$:      .WORD   20$-3$,-                                                    :
                                                                        21$-3$,-                                                    :
                                                                        22$-3$                                                      :
                        03      08   AE  E9 00056          BLBC    QID, 4$                                                     : 1494
                                     00DA 31 0005A          BRW     19$                                                         :
                03      04      AC      04  E0 0005D 4$:   BBS     #4, QSM, 5$                                                 : 1500
                                     0082 31 00062          BRW     12$                                                         :
                                        5A  D5 00065 5$:    TSTL    SMQ_N                                                       :
                                        0F  12 00067        BNEQ    6$                                                          :
                        5A              01  D0 00069        MOVL    #1, SMQ_N                                                   : 1502
                                        01  DD 0006C        PUSHL   #1                                                          :
                00000000G       EF      01  FB 0006E        CALLS   #1, READ_RECORD                                             :
                        54              50  D0 00075        MOVL    R0, SMQ                                                     :
                        53              5A  D0 00078 6$:    MOVL    SMQ_N, SMQ_NP                                               : 1510
                                        5A  D4 0007B        CLRL    SMQ_N                                                       : 1511
                                        5B  D5 0007D        TSTL    SQX_N                                                       : 1516
                                        07  12 0007F        BNEQ    8$                                                          :
                        5B      64      A4  D0 00081        MOVL    100(SMQ), SQX_N                                             : 1519
                        6E              1C  CE 00085 7$:    MNEGL   #28, SQE_N                                                  : 1520
                                        5B  D5 00088 8$:    TSTL    SQX_N                                                       : 1526
                                        6B  13 0008A        BEQL    14$                                                         :
                                04      AE  D5 0008C        TSTL    SQX                                                         : 1534
                                        0D  12 0008F        BNEQ    9$                                                          :
                                        5B  DD 00091        PUSHL   SQX_N                                                       :
                00000000G       EF      01  FB 00093        CALLS   #1, READ_RECORD                                             :
                        04      AE      50  D0 0009A        MOVL    R0, SQX                                                     :
                        6E              28  C0 0009E 9$:    ADDL2   #40, SQE_N                                                  : 1548
                000001EC        8F      6E  D1 000A1        CMPL    SQE_N, #492                                                 : 1549
                                        28  1E 000A8        BGEQU   11$                                                         :
                50      04      AE      6E  C1 000AA        ADDL3   SQE_N, SQX, SQE                                             : 1551
                                        60  95 000AF        TSTB    (SQE)                                                       : 1552
                                        1F  13 000B1        BEQL    11$                                                         :
                        08      AC  24  A0  D1 000B3        CMPL    36(SQE), SMQ_NF                                             : 1557
                                        12  13 000B8        BEQL    10$                                                         :
                                51  0C  AC  D0 000BA        MOVL    SMQ_F, R1                                                   : 1558
                52      0C      A1  20  A0  8D 000BE        XORB3   32(SQE), 12(R1), R2                                         :
                                D7      52  E8 000C4        BLBS    R2, 9$                                                      :
                D2      20      A0      01  E1 000C7        BBC     #1, 32(SQE), 9$                                             : 1559
                                5A      24  A0  D0 000CC 10$:   MOVL    36(SQE), SMQ_N                                              : 1562
                                        25  11 000D0        BRB     14$                                                         : 1563
                                52      04  BE  D0 000D2 11$:   MOVL    @SQX, SQX_NS                                                : 1570
                                        5B  DD 000D6        PUSHL   SQX_N                                                       : 1571
                        00000000G       EF      01  FB 000D8        CALLS   #1, RELEASE_RECORD                                          :
                                04      AE  D4 000DF        CLRL    SQX                                                         : 1572
                                5B      52  D0 000E2        MOVL    SQX_NS, SQX_N                                               : 1573
                                        9E  11 000E5        BRB     7$                                                          : 1574
                                53      5A  D0 000E7 12$:   MOVL    SMQ_N, SMQ_NP                                               : 1583
                                01      5A  D1 000EA        CMPL    SMQ_N, #1                                                   : 1584
                                        06  1A 000ED        BGTRU   13$                                                         :
                                5A      08  AC  D0 000EF        MOVL    SMQ_NF, SMQ_N                                               : 1585
                                        02  11 000F3        BRB     14$                                                         :
                                        5A  D4 000F5 13$:   CLRL    SMQ_N                                                       : 1586
                                        53  D5 000F7 14$:   TSTL    SMQ_NP                                                      : 1593
                                        09  13 000F9        BEQL    15$                                                         :
                                        53  DD 000FB        PUSHL   SMQ_NP                                                      :
                        00000000G       EF      01  FB 000FD        CALLS   #1, RELEASE_RECORD                                          :
                                        5A  D5 00104 15$:   TSTL    SMQ_N                                                       : 1594
```

```
                                    0A  12 00106          BNEQ    16$
                          50 00048040   8F  D0 00108          MOVL    #294976, R0
                                02F6    31  00010F          BRW     66$
                                    5A  DD 00112 16$:       PUSHL   SMQ_N                                   ;1595
                    00000000G       EF  01 FB 00114         CALLS   #1, READ_RECORD
                                    54  50 D0 00118         MOVL    R0, SMQ
                03          04      AC  03 E0 0011E         BBS     #3, QSM, 18$                            ;1601
                                FF22    31  00123 17$:      BRW     2$
                            5A      08  AC  D1 00126 18$:    CMPL    SMQ_NF, SMQ_N                           ;1602
                                    F7  12 0012A            BNEQ    17$
                                78      A4  D5 0012C        TSTL    120(SMQ)                                ;1603
                                    F2  13 0012F            BEQL    17$
                            56      78  8F  9A 00131        MOVZBL  #120, LIST_OFFSET                       ;1606
                                    67  11 00135            BRB     24$                                     ;1605
                E7          04      AC  04 E1 00137 19$:    BBC     #4, QSM, 17$                            ;1616
                            50      0C  AC  D0 0013C        MOVL    SMQ_F, R0                               ;1617
                51          0C      A4  0C  A0 8D 00140     XORB3   12(R0), 12(SMQ), R1
                                    DA  51  E8 00146        BLBS    R1, 17$
                                48      A4  D5 00149        TSTL    72(SMQ)                                 ;1618
                                    D5  13 0014C            BEQL    17$
                            56      48  8F  9A 0014E        MOVZBL  #72, LIST_OFFSET                        ;1621
                                    4A  11 00152            BRB     24$                                     ;1620
                            CB      04  AC  E9 00154 20$:   BLBC    QSM, 17$                                ;1444
                            50      0C  AC  D0 00158        MOVL    SMQ_F, R0                               ;1445
                                0100    C0  B5 0015C        TSTW    256(R0)
                                    C1  13 00160            BEQL    17$
                            56      4C  8F  9A 00162        MOVZBL  #76, LIST_OFFSET                        ;1448
                                    36  11 00166            BRB     24$                                     ;1447
                B6          04      AC  01 E1 00168 21$:    BBC     #1, QSM, 17$                            ;1456
                            50      0C  AC  D0 0016D        MOVL    SMQ_F, R0                               ;1457
                                010C    C0  B5 00171        TSTW    268(R0)
                                    AC  13 00175            BEQL    17$
                            56      68  8F  9A 00177        MOVZBL  #104, LIST_OFFSET                       ;1460
                                    21  11 0017B            BRB     24$                                     ;1459
                A1          04      AC  02 E1 0017D 22$:    BBC     #2, QSM, 17$                            ;1468
                            50      0C  AC  D0 00182        MOVL    SMQ_F, R0                               ;1469
                                0102    C0  B5 00186        TSTW    258(R0)
                                    97  13 0018A            BEQL    17$
                            50      0C  AC  D0 0018C        MOVL    SMQ_F, R0                               ;1472
                            06      0C  A0  E9 00190        BLBC    12(R0), 23$
                            56      54  8F  9A 00194        MOVZBL  #84, LIST_OFFSET                        ;1473
                                    04  11 00198            BRB     24$
                            56      5C  8F  9A 0019A 23$:   MOVZBL  #92, LIST_OFFSET                        ;1474
                                    5A  D5 0019E 24$:       TSTL    SMQ_N                                   ;1632
                                    0F  12 001A0            BNEQ    25$
                            5A          01  D0 001A2        MOVL    #1, SMQ_N
                                        01  DD 001A5        PUSHL   #1
                    00000000G           EF  01 FB 001A7     CALLS   #1, READ_RECORD
                                        54  50 D0 001AE     MOVL    R0, SMQ
                            59          5A  D0 001B1 25$:   MOVL    SMQ_N, SJH_NP                           ;1633
                                        57  D4 001B4        CLRL    SJH_P                                   ;1634
                                        04  11 001B6        BRB     27$                                     ;1635
                                        57  D5 001B8 26$:   TSTL    SJH_P                                   ;1638
                                        08  12 001BA        BNEQ    28$
                                    6644    9F  001BC 27$:  PUSHAB  (LIST_OFFSET)[SMQ]                      ;1639
                            58              9E  D0 001BF    MOVL    @(SP)+, SJH_N
                                            03  11 001C2    BRB     29$
```

```
                                58          67  D0 001C4 28$:   MOVL    (SJH_P), SJH_N                          ; 1640
                                            03  12 001C7 29$:   BNEQ    30$                                     ; 1645
                                          0229  31 001C9         BRW     64$
                                            58  DD 001CC 30$:   PUSHL   SJH_N                                   ; 1647
                    00000000G   EF              01  FB 001CE         CALLS   #1, READ_RECORD
                                55              50  D0 001D5         MOVL    R0, SJH
                          0134  C5          08  AC  D1 001D8         CMPL    SMQ_NF, 308(SJH)                    ; 1650
                                            13  13 001DE         BEQL    33$
                                03          08  AE  D1 001E0         CMPL    QID, #3                             ; 1651
                                            03  1A 001E4         BGTRU   32$
                                          01F2  31 001E6 31$:     BRW     62$
                                F9          08  AE  E9 001E9 32$:   BLBC    QID, 31$
                                5A          08  AC  D1 001ED         CMPL    SMQ_NF, SMQ_N
                                            F3  12 001F1         BNEQ    31$
                                            10  AC  D5 001F3 33$:   TSTL    ENTRY                               ; 1655
                                            09  13 001F6         BEQL    34$
    08  A5      10  BC                  10          00  ED 001F8         CMPZV   #0, #16, @ENTRY, 8(SJH)         ; 1657
                                            27  11 001FF         BRB     35$
                                50          14  AC  D0 00201 34$:   MOVL    JOBNAME, R0                         ; 1659
                                            23  13 00205         BEQL    36$
                                51      0108    C5  9A 00207         MOVZBL  264(SJH), R1                        ; 1663
                                60              51  B1 0020C         CMPW    R1, (R0)
                                            D5  12 0020F         BNEQ    31$
                0109    C5      02  B0          60  29 00211         CMPC3   (R0), @2(R0), 265(SJH)              ; 1665
                                            CC  12 00218         BNEQ    31$
                                50 00000000'    EF  D0 0021A         MOVL    MBX, R0                             ; 1669
                0148    C5      10  A0          0C  29 00221         CMPC3   #12, 16(R0), 328(SJH)               ; 1670
                                            BC  12 00228 35$:   BNEQ    31$
                                        18  AC  D5 0022A 36$:   TSTL    ACCESS_CHECK                        ; 1682
                                            4B  13 0022D         BEQL    40$
                                            55  DD 0022F         PUSHL   SJH                                 ; 1684
                                        0C  AC  DD 00231         PUSHL   SMQ_F
                        18  BC              02  FB 00234         CALLS   #2, @ACCESS_CHECK
                                3F              50  E8 00238         BLBS    R0, 40$
                                            5B  D5 0023B         TSTL    SQX_N                               ; 1687
                                            09  13 0023D         BEQL    37$
                                            5B  DD 0023F         PUSHL   SQX_N                               ; 1688
                00000000G   EF              01  FB 00241         CALLS   #1, RELEASE_RECORD
                                            5A  D5 00248 37$:   TSTL    SMQ_N                               ; 1689
                                            0E  13 0024A         BEQL    38$
                                59              5A  D1 0024C         CMPL    SMQ_N, SJH_NP
                                            09  13 0024F         BEQL    38$
                                            5A  DD 00251         PUSHL   SMQ_N                               ; 1690
                00000000G   EF              01  FB 00253         CALLS   #1, RELEASE_RECORD
                                            59  D5 0025A 38$:   TSTL    SJH_NP                              ; 1691
                                            09  13 0025C         BEQL    39$
                                            59  DD 0025E         PUSHL   SJH_NP                              ; 1692
                00000000G   EF              01  FB 00260         CALLS   #1, RELEASE_RECORD
                                            58  DD 00267 39$:   PUSHL   SJH_N                               ; 1693
                00000000G   EF              01  FB 00269         CALLS   #1, RELEASE_RECORD
                                50 00048020     8F  D0 00270         MOVL    #294944, R0                         ; 1694
                                          018E  31 00277         BRW     66$
                                53              01  D0 0027A 40$:   MOVL    #1, REMOVING                        ; 1701
                    02          00      1C  AC  CF 0027D         CASEL   REMOVE, #0, #2                      ; 1702
                  0008        000F            000D      00282 41$:   .WORD   43$-41$,-
                                                                              44$-41$,-
                                                                              42$-41$
```

```
                                      05  11 00288        BRB     43$                                ; 1705
        02        10   A5             03  E1 0028A 42$:   BBC     #3, 16(SJH), 44$                   ; 1709
                                      53  D4 0028F 43$:   CLRL    REMOVING                           ;
                       03             53  E8 00291 44$:   BLBS    REMOVING, 45$                      ; 1713
                                    00F1  31 00294        BRW     57$                                ;
                       03        08   AE  D1 00297 45$:   CMPL    QID, #3                            ; 1720
                                      3D  1A 0029B        BGTRU   51$                                ;
        02             01        08   AE  CF 0029D        CASEL   QID, #1, #2                        ; 1723
      001C           0012           0008     002A2 46$:   .WORD   47$-46$,-                          ;
                                                                  48$-46$,-                          ;
                                                                  49$-46$                            ;
                                      1C  11 002A8        BRB     50$                                ;
                       50        0C   AC  D0 002AA 47$:   MOVL    SMQ_F, R0                          ; 1730
                               0100   C0  B7 002AE        DECW    256(R0)                            ; 1731
                                      12  11 002B2        BRB     50$                                ; 1730
                       50        0C   AC  D0 002B4 48$:   MOVL    SMQ_F, R0                          ; 1734
                               010C   C0  B7 002B8        DECW    268(R0)                            ; 1735
                                      08  11 002BC        BRB     50$                                ; 1734
                       50        0C   AC  D0 002BE 49$:   MOVL    SMQ_F, R0                          ; 1738
                               0102   C0  B7 002C2        DECW    258(R0)                            ; 1739
                                 08   AC  DD 002C6 50$:   PUSHL   SMQ_NF                             ; 1742
           00000000G   EF             01  FB 002C9        CALLS   #1, READ_RECORD                    ;
                                 08   AC  DD 002D0        PUSHL   SMQ_NF                             ; 1743
           00000000G   EF             01  FB 002D3        CALLS   #1, REWRITE_RECORD                 ;
                                      55  DD 002DA 51$:   PUSHL   SJH                                ; 1749
                                      58  DD 002DC        PUSHL   SJH_N                              ;
           00000000G   EF             02  FB 002DE        CALLS   #2, UPDATE_GETQUI_DATA             ;
                                      57  D5 002E5        TSTL    SJH_P                              ; 1750
                                      6D  12 002E7        BNEQ    54$                                ;
                                    6644  9F 002E9        PUSHAB  (LIST_OFFSET)[SMQ]                 ; 1753
                       9E             65  D0 002EC        MOVL    (SJH), @(SP)+                      ;
                                      06  12 002EF        BNEQ    52$                                ; 1754
                                 04 A644  9F 002F1        PUSHAB  4(LIST_OFFSET)[SMQ]                ; 1755
                                      9E  D4 002F5        CLRL    @(SP)+                             ;
                                      5A  DD 002F7 52$:   PUSHL   SMQ_N                              ; 1756
           00000000G   EF             01  FB 002F9        CALLS   #1, READ_RECORD                    ;
                                      5A  DD 00300        PUSHL   SMQ_N                              ; 1757
           00000000G   EF             01  FB 00302        CALLS   #1, REWRITE_RECORD                 ;
                       02        08   AE  D1 00309        CMPL    QID, #2                            ; 1758
                                      77  12 0030D        BNEQ    56$                                ;
                       7E             01  7D 0030F        MOVQ    #1, -(SP)                          ; 1766
           00000000G   00             02  FB 00312        CALLS   #2, SYS$CANTIM                     ;
                                      65  D5 00319        TSTL    (SJH)                              ; 1767
                                      79  13 0031B        BEQL    59$                                ;
                       52             65  D0 0031D        MOVL    (SJH), SJH_N2                      ; 1770
                                      52  DD 00320        PUSHL   SJH_N2                             ;
           00000000G   EF             01  FB 00322        CALLS   #1, READ_RECORD                    ;
                                      01  DD 00329        PUSHL   #1                                 ; 1774
                          00000000G   EF  9F 0032B        PUSHAB  AFTER_AST                          ;
                               0098   C0  9F 00331        PUSHAB  152(SJH_2)                         ;
                                      7E  D4 00335        CLRL    -(SP)                              ;
           00000000G   00             04  FB 00337        CALLS   #4, SYS$SETIMR                     ;
                       11             50  E8 0033E        BLBS    STATUS, 53$                        ; 1775
                                      50  DD 00341        PUSHL   STATUS                             ; 1777
                                      7E  D4 00343        CLRL    -(SP)                              ;
                          0004845A    8F  DD 00345        PUSHL   #296026                            ;
           00000000G   00             03  FB 0034B        CALLS   #3, LIB$SIGNAL                     ;
```

```
                                    52  DD 00352 53$:    PUSHL   SJH_N2                          ; 1778
                                    39  11 00354         BRB     58$
                                    65  D0 00356 54$:    MOVL    (SJH), (SJH_P)                 ; 1784
                                    19  12 00359         BNEQ    55$                            ; 1785
                                04 A644 9F 0035B         PUSHAB  4(LIST_OFFSET)[SMQ]            ; 1788
                                    59  D0 0035F         MOVL    SJH_NP, @(SP)+
                                    5A  DD 00362         PUSHL   SMQ_N                          ; 1789
                00000000G       EF      01  FB 00364     CALLS   #1, READ_RECORD
                                    5A  DD 00368         PUSHL   SMQ_N                          ; 1790
                00000000G       EF      01  FB 0036D     CALLS   #1, REWRITE_RECORD
                                    59  DD 00374 55$:    PUSHL   SJH_NP                         ; 1792
                00000000G       EF      01  FB 00376     CALLS   #1, READ_RECORD
                                    59  DD 0037D         PUSHL   SJH_NP                         ; 1793
                00000000G       EF      01  FB 0037F     CALLS   #1, REWRITE_RECORD
                                    0E  11 00386 56$:    BRB     59$                            ; 1713
                            5A          59  D1 00388 57$:    CMPL    SJH_NP, SMQ_N              ; 1798
                                    09  13 0038B         BEQL    59$
                                    59  DD 0038D         PUSHL   SJH_NP
                00000000G       EF      01  FB 0038F 58$:    CALLS   #1, RELEASE_RECORD
                            08          6C  91 00396 59$:    CMPB    (AP), #8                   ; 1802
                                    3B  1F 00399         BLSSU   61$
                                20  AC  D5 0039B         TSTL    32(AP)
                                    36  13 0039E         BEQL    61$
                            50      20  AC  D0 003A0     MOVL    CTX, R0                        ; 1805
                            60          56  D0 003A4     MOVL    LIST_OFFSET, (R0)
                        04  A0      08  AE  D0 003A7     MOVL    QID, 4(R0)                     ; 1806
                        08  A0          5B  D0 003AC     MOVL    SQX_N, 8(R0)                   ; 1807
                        0C  A0      04  AE  D0 003B0     MOVL    SQX, 12(R0)                    ; 1808
                        10  A0          6E  D0 003B5     MOVL    SQE_N, 16(R0)                  ; 1809
                        14  A0          5A  D0 003B9     MOVL    SMQ_N, 20(R0)                  ; 1810
                        18  A0          54  D0 003BD     MOVL    SMQ, 24(R0)                    ; 1811
                            0A          53  E9 003C1     BLBC    REMOVING, 60$                  ; 1816
                        1C  A0          59  D0 003C4     MOVL    SJH_NP, 28(R0)                 ; 1815
                        20  A0          57  D0 003C8     MOVL    SJH_P, 32(R0)                  ; 1816
                                    08  11 003CC         BRB     61$                            ; 1812
                        1C  A0          58  D0 003CE 60$:    MOVL    SJH_N, 28(R0)              ; 1820
                        20  A0          55  D0 003D2     MOVL    SJH, 32(R0)                    ; 1821
                            50          01  D0 003D6 61$:    MOVL    #1, R0                     ; 1826
                                    2D  11 003D9         BRB     66$
                            5A          59  D1 003DB 62$:    CMPL    SJH_NP, SMQ_N              ; 1830
                                    09  13 003DE         BEQL    63$
                                    59  DD 003E0         PUSHL   SJH_NP
                00000000G       EF      01  FB 003E2     CALLS   #1, RELEASE_RECORD
                            59          58  D0 003E9 63$:    MOVL    SJH_N, SJH_NP              ; 1831
                            57          55  D0 003EC     MOVL    SJH, SJH_P                     ; 1832
                            58          65  D0 003EF     MOVL    (SJH), SJH_N                   ; 1833
                                    FDD2  31  003F2     BRW     29$                            ; 1645
                            5A          59  D1 003F5 64$:    CMPL    SJH_NP, SMQ_N              ; 1839
                                    09  13 003F8         BEQL    65$
                                    59  DD 003FA         PUSHL   SJH_NP
                00000000G       EF      01  FB 003FC     CALLS   #1, RELEASE_RECORD
                                    56  D4 00403 65$:    CLRL    LIST_OFFSET                    ; 1840
                                    FC39  31  00405     BRW     1$                             ; 1415
                            5B          54  D0 00408 66$:    MOVL    R4, R11                    ; 1843
                            59          55  D0 0040B     MOVL    R5, R9
                                        04  0040E         RET                                    ;
```

; Routine Size: 1039 bytes,   Routine Base: CODE + 00BD

```
;  809     1844  1 GLOBAL ROUTINE DEQUEUE_OPEN_JOB(SJH_N;SJH): L_OUTPUT_1=
;  810     1845  1
;  811     1846  1 !++
;  812     1847  1 !
;  813     1848  1 ! FUNCTIONAL DESCRIPTION:
;  814     1849  1 !       This routine searches the open job queue for a specified job, and
;  815     1850  1 !       dequeues the job.
;  816     1851  1 !
;  817     1852  1 ! INPUT PARAMETERS:
;  818     1853  1 !       SJH_N           - Record number of SJH record.
;  819     1854  1 !
;  820     1855  1 ! IMPLICIT INPUTS:
;  821     1856  1 !       NONE
;  822     1857  1 !
;  823     1858  1 ! OUTPUT PARAMETERS:
;  824     1859  1 !       SJH             - Pointer to SJH record.
;  825     1860  1 !
;  826     1861  1 ! IMPLICIT OUTPUTS:
;  827     1862  1 !       NONE
;  828     1863  1 !
;  829     1864  1 ! ROUTINE VALUE:
;  830     1865  1 !       TRUE            - Job found.
;  831     1866  1 !       FALSE           - Job not found.
;  832     1867  1 !
;  833     1868  1 ! SIDE EFFECTS:
;  834     1869  1 !       NONE
;  835     1870  1 !
;  836     1871  1 !--
;  837     1872  1
;  838     1873  2 BEGIN
;  839     1874  2 MAP
;  840     1875  2         SJH:            REF BBLOCK;     ! Pointer to SJH
;  841     1876  2 LOCAL
;  842     1877  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  843     1878  2         SMQ_N,                          ! Record number of SMQ
;  844     1879  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  845     1880  2         SJH_NP,                         ! Record number of predecessor of SJH
;  846     1881  2         SJH_P:          REF BBLOCK,     ! Pointer to predecessor of SJH
;  847     1882  2         SJH_NT;                         ! Record number of SJH
;  848     1883  2
;  849     1884  2
;  850     1885  2 SQH = READ_RECORD(SJH_NP = SQH$K_RECNO);
;  851     1886  2 SJH_NT = .SQH[SQH$L_OPEN_LIST];
;  852     1887  2 WHILE .SJH_NT NEQ 0 DO
;  853     1888  3     BEGIN
;  854     1889  3     SJH = READ_RECORD(.SJH_NT);
;  855     1890  3     IF .SJH_N EQL .SJH_NT
;  856     1891  3     THEN
;  857     1892  4         BEGIN
;  858     1893  4         IF .SJH_NP EQL SQH$K_RECNO
;  859     1894  4         THEN
;  860     1895  5             BEGIN
;  861     1896  5             SQH[SQH$L_OPEN_LIST] = .SJH[SYM$L_LINK];
;  862     1897  5             IF .SJH[SYM$L_LINK] EQL 0 THEN SQH[SQH$L_OPEN_LIST_END] = 0;
;  863     1898  5             REWRITE_RECORD(SQH$K_RECNO);
;  864     1899  5             END
;  865     1900  4         ELSE
```

```
;  866          1901  5              BEGIN
;  867          1902  5              SJH_P[SYM$L_LINK] = .SJH[SYM$L_LINK];
;  868          1903  5              IF .SJH[SYM$L_LINK] EQL 0 THEN SQH[SQH$L_OPEN_LIST_END] = .SJH_NP;
;  869          1904  5              REWRITE_RECORD(.SJH_NP);
;  870          1905  5              REWRITE_RECORD(SQH$K_RECNO);
;  871          1906  4              END;
;  872          1907  4          SMQ_N = .SJH[SJH$L_QUEUE_LINK];
;  873          1908  4          IF .SMQ_N NEQ 0
;  874          1909  4          THEN
;  875          1910  5              BEGIN
;  876          1911  5              ! Queue pointer is OK, update queue record.
;  877          1912  5              !
;  878          1913  5              SMQ = READ_RECORD(.SMQ_N);
;  879          1914  5              SMQ[SMQ$W_OPEN_JOB_COUNT] = .SMQ[SMQ$W_OPEN_JOB_COUNT] - 1;
;  880          1915  5              REWRITE_RECORD(.SMQ_N);
;  881          1916  4              END;
;  882          1917  4          RETURN TRUE;
;  883          1918  3          END;
;  884          1919  3      IF .SJH_NP NEQ SQH$K_RECNO THEN RELEASE_RECORD(.SJH_NP);
;  885          1920  3      SJH_NP = .SJH_NT;
;  886          1921  3      SJH_P = .SJH;
;  887          1922  3      SJH_NT = .SJH[SYM$L_LINK];
;  888          1923  2      END;
;  889          1924  2 IF .SJH_NP NEQ SQH$K_RECNO THEN RELEASE_RECORD(.SJH_NP);
;  890          1925  2 RELEASE_RECORD(SQH$K_RECNO);
;  891          1926  2 FALSE
;  892          1927  1 END;
; INFO#250                L1:1902
; Referenced LOCAL symbol SJH_P is probably not initialized
```

```
                                07FC 00000          .ENTRY  DEQUEUE_OPEN_JOB, Save R2,R3,R4,R5,R6,R7,-      ; 1844
                                                            R8,R9,R10
                5A 00000000G    EF   9E 00002          MOVAB   READ_RECORD, R10                             ;
                59 00000000G    EF   9E 00009          MOVAB   RELEASE_RECORD, R9                           ;
                58 00000000G    EF   9E 00010          MOVAB   REWRITE_RECORD, R8                           ;
                55              01   D0 00017          MOVL    #1, SJH_NP                                   ; 1885
                                01   DD 0001A          PUSHL   #1                                           ;
                6A              01   FB 0001C          CALLS   #1, READ_RECORD                              ;
                53              50   D0 0001F          MOVL    R0, SQH                                      ;
                54        4C    A3   D0 00022          MOVL    76(SQH), SJH_NT                              ; 1886
                                62   13 00026 1$:      BEQL    8$                                           ; 1887
                                54   DD 00028          PUSHL   SJH_NT                                       ; 1889
                6A              01   FB 0002A          CALLS   #1, READ_RECORD                              ;
                5B              50   D0 0002D          MOVL    R0, SJH                                      ;
                54        04    AC   D1 00030          CMPL    SJH_N, SJH_NT                                ; 1890
                                3F   12 00034          BNEQ    6$                                           ;
                01              55   D1 00036          CMPL    SJH_NP, #1                                   ; 1893
                                0B   12 00039          BNEQ    2$                                           ;
          4C    A3              6B   D0 0003B          MOVL    (SJH), 76(SQH)                               ; 1896
                                13   12 0003F          BNEQ    4$                                           ; 1897
                          50    A3   D4 00041          CLRL    80(SQH)                                      ;
                                0E   11 00044          BRB     4$                                           ; 1898
                67              6B   D0 00046 2$:      MOVL    (SJH), (SJH_P)                               ; 1902
```

```
                                 04  12 00049        BNEQ    3$                                   ; 1903
                   50   A3       55  D0 0004B        MOVL    SJH_NP, 80(SQH)                      ;
                                 55  DD 0004F 3$:    PUSHL   SJH_NP                               ; 1904
                        68       01  FB 00051        CALLS   #1, REWRITE_RECORD                   ;
                                 01  DD 00054 4$:    PUSHL   #1                                   ; 1905
                        68       01  FB 00056        CALLS   #1, REWRITE_RECORD                   ;
                        56  0134 CB  D0 00059        MOVL    308(SJH), SMQ_N                      ; 1907
                                 11  13 0005E        BEQL    5$                                   ; 1908
                                 56  DD 00060        PUSHL   SMQ_N                                ; 1913
                        6A       01  FB 00062        CALLS   #1, READ_RECORD                      ;
                        52       50  D0 00065        MOVL    R0, SMQ                              ;
                            0100 C2  B7 00068        DECW    256(SMQ)                             ; 1914
                                 56  DD 0006C        PUSHL   SMQ_N                                ; 1915
                        68       01  FB 0006E        CALLS   #1, REWRITE_RECORD                   ;
                        50       01  D0 00071 5$:    MOVL    #1, R0                               ; 1917
                                     04 00074        RET                                          ;
                        01       55  D1 00075 6$:    CMPL    SJH_NP, #1                           ; 1919
                                 05  13 00078        BEQL    7$                                   ;
                                 55  DD 0007A        PUSHL   SJH_NP                               ;
                        69       01  FB 0007C        CALLS   #1, RELEASE_RECORD                   ;
                        55       54  D0 0007F 7$:    MOVL    SJH_NT, SJH_NP                       ; 1920
                        57       5B  D0 00082        MOVL    SJH, SJH_P                           ; 1921
                        54       6B  D0 00085        MOVL    (SJH), SJH_NT                        ; 1922
                                 9C  11 00088        BRB     1$                                   ; 1887
                        01       55  D1 0008A 8$:    CMPL    SJH_NP, #1                           ; 1924
                                 05  13 0008D        BEQL    9$                                   ;
                                 55  DD 0008F        PUSHL   SJH_NP                               ;
                        69       01  FB 00091        CALLS   #1, RELEASE_RECORD                   ;
                                 01  DD 00094 9$:    PUSHL   #1                                   ; 1925
                        69       01  FB 00096        CALLS   #1, RELEASE_RECORD                   ;
                                 50  D4 00099        CLRL    R0                                   ; 1927
                                     04 0009B        RET                                          ;

; Routine Size:  156 bytes,    Routine Base:  CODE + 04CC
```

```
;   894          1928  1 GLOBAL ROUTINE ALLOCATE_ENTRY_NUMBER(P_ENTRY_NUMBER)=
;   895          1929  1
;   896          1930  1 !++
;   897          1931  1 !
;   898          1932  1 ! FUNCTIONAL DESCRIPTION:
;   899          1933  1 !       This routine allocates a new job entry number.
;   900          1934  1 !
;   901          1935  1 ! INPUT PARAMETERS:
;   902          1936  1 !       P_ENTRY_NUMBER  - Address of a longword to receive the entry number.
;   903          1937  1 !
;   904          1938  1 ! IMPLICIT INPUTS:
;   905          1939  1 !       NONE
;   906          1940  1 !
;   907          1941  1 ! OUTPUT PARAMETERS:
;   908          1942  1 !       NONE
;   909          1943  1 !
;   910          1944  1 ! IMPLICIT OUTPUTS:
;   911          1945  1 !       NONE
;   912          1946  1 !
;   913          1947  1 ! ROUTINE VALUE:
;   914          1948  1 !       Completion status.
;   915          1949  1 !
;   916          1950  1 ! SIDE EFFECTS:
;   917          1951  1 !       NONE
;   918          1952  1 !
;   919          1953  1 !--
;   920          1954  1
;   921          1955  2 BEGIN
;   922          1956  2 LOCAL
;   923          1957  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;   924          1958  2         SEB_N,                          ! Record number of bitmap extension
;   925          1959  2         SEB:            REF BBLOCK,     ! Pointer to bitmap extension
;   926          1960  2         ENTRY_NUMBER,                   ! Trial entry number
;   927          1961  2         ENTRY_NUMBER_LIMIT,             ! Limit for entry number loop
;   928          1962  2         BIT_NUMBER,                     ! Bit number within bitmap block
;   929          1963  2         BLOCK_NUMBER,                   ! Offset to bitmap extension block
;   930          1964  2         Q:              VECTOR[2],      ! Temporary for EDIV
;   931          1965  2         STATUS;                         ! Status return
;   932          1966  2
;   933          1967  2
;   934          1968  2 ! Read the queue header.
;   935          1969  2 !
;   936          1970  2 SQH = READ_RECORD(SQH$K_RECNO);
;   937          1971  2 SEB_N = 0;
;   938          1972  2
;   939          1973  2
;   940          1974  2 ! Search the portion of the bitmap from NEXT_ENTRY_NUMBER to the end and
;   941          1975  2 ! then the portion of the bitmap from the beginning to NEXT_ENTRY_NUMBER.
;   942          1976  2 !
;   943          1977  2 ENTRY_NUMBER = .SQH[SQH$L_NEXT_ENTRY_NUMBER];
;   944          1978  2 ENTRY_NUMBER_LIMIT = .SQH[SQH$L_HIGHEST_ENTRY_NUMBER];
;   945          1979  2 DECR I FROM 1 TO 0 DO
;   946          1980  3     BEGIN
;   947          1981  3
;   948          1982  3     ! Search the specified portion of the bitmap.
;   949          1983  3     !
;   950          1984  3     WHILE .ENTRY_NUMBER LEQU .ENTRY_NUMBER_LIMIT DO
```

```
;  951          1985  4          BEGIN
;  952          1986  4
;  953          1987  4          ! Normalize the entry number to the bit number.
;  954          1988  4          !
;  955          1989  4          BIT_NUMBER = .ENTRY_NUMBER - 1;
;  956          1990  4          IF .BIT_NUMBER LSSU SQH$S_ENTRY_BITMAP * 8
;  957          1991  4          THEN
;  958          1992  5              BEGIN
;  959          1993  5
;  960          1994  5              ! The bit is not in an extension record.  Try to allocate the
;  961          1995  5              ! specified bit in the queue header.  If this succeeds, update
;  962          1996  5              ! data structures and return success.
;  963          1997  5              !
;  964          1998  5              IF TESTBITCS(BITVECTOR[SQH[SQH$B_ENTRY_BITMAP], .BIT_NUMBER])
;  965          1999  5              THEN
;  966          2000  6                  BEGIN
;  967          2001  6                  SQH[SQH$L_NEXT_ENTRY_NUMBER] = .ENTRY_NUMBER + 1;
;  968          2002  6                  .P_ENTRY_NUMBER = .ENTRY_NUMBER;
;  969          2003  6                  IF .SEB_N NEQ 0 THEN RELEASE_RECORD(.SEB_N);
;  970          2004  6                  REWRITE_RECORD(SQH$K_RECNO);
;  971          2005  6                  RETURN SS$_NORMAL;
;  972          2006  5                  END;
;  973          2007  5              END
;  974          2008  4          ELSE
;  975          2009  5              BEGIN
;  976          2010  5
;  977          2011  5              ! The bit is in an extension record.  Determine the index within
;  978          2012  5              ! the extension record vector, and the bit number within the
;  979          2013  5              ! selected extension record.
;  980          2014  5              !
;  981          2015  5              Q[0] = .BIT_NUMBER - SQH$S_ENTRY_BITMAP * 8;
;  982          2016  5              Q[1] = 0;
;  983          2017  5              EDIV(%REF(SYM$S_DATA * 8), Q, BLOCK_NUMBER, BIT_NUMBER);
;  984          2018  5
;  985          2019  5
;  986          2020  5              ! If the wrong extension record (or no extension record) is in
;  987          2021  5              ! memory, read the required record.
;  988          2022  5              !
;  989          2023  5              IF .SEB_N NEQ .VECTOR[SQH[SQH$L_ENTRY_BITMAP_VECTOR], .BLOCK_NUMBER]
;  990          2024  5              THEN
;  991          2025  6                  BEGIN
;  992          2026  6                  IF .SEB_N NEQ 0 THEN RELEASE_RECORD(.SEB_N);
;  993          2027  6                  SEB_N = .VECTOR[SQH[SQH$L_ENTRY_BITMAP_VECTOR], .BLOCK_NUMBER];
;  994          2028  6                  SEB = READ_RECORD(.SEB_N);
;  995          2029  5                  END;
;  996          2030  5
;  997          2031  5
;  998          2032  5              ! Try to allocate the specified bit in the extension record.  If
;  999          2033  5              ! this succeeds, update data structures and return success.
; 1000          2034  5              !
; 1001          2035  5              IF TESTBITCS(BITVECTOR[SEB[SYM$T_DATA], .BIT_NUMBER])
; 1002          2036  5              THEN
; 1003          2037  6                  BEGIN
; 1004          2038  6                  SQH[SQH$L_NEXT_ENTRY_NUMBER] = .ENTRY_NUMBER + 1;
; 1005          2039  6                  .P_ENTRY_NUMBER = .ENTRY_NUMBER;
; 1006          2040  6                  REWRITE_RECORD(.SEB_N);
; 1007          2041  6                  REWRITE_RECORD(SQH$K_RECNO);
```

```
;  1008     2042  6                    RETURN SS$_NORMAL;
;  1009     2043  5                    END;
;  1010     2044  4                END;
;  1011     2045  4
;  1012     2046  4
;  1013     2047  4            ! Advance to the next entry number.
;  1014     2048  4            !
;  1015     2049  4            ENTRY_NUMBER = .ENTRY_NUMBER + 1;
;  1016     2050  3            END;
;  1017     2051  3
;  1018     2052  3
;  1019     2053  3        ! Set up to search the beginning of the bitmap.
;  1020     2054  3        !
;  1021     2055  3        ENTRY_NUMBER = 1;
;  1022     2056  3        ENTRY_NUMBER_LIMIT = .SQH[SQH$L_NEXT_ENTRY_NUMBER] - 1;
;  1023     2057  2        END;
;  1024     2058  2
;  1025     2059  2
;  1026     2060  2    ! All bits were set in the existing blocks (extremely unlikely).  Determine the
;  1027     2061  2    ! offset within the vector of a new block.
;  1028     2062  2    !
;  1029     2063  2    IF .SEB_N NEQ 0 THEN RELEASE_RECORD(.SEB_N);
;  1030     2064  2    BLOCK_NUMBER =
;  1031     2065  2        (.SQH[SQH$L_HIGHEST_ENTRY_NUMBER] - SQH$S_ENTRY_BITMAP * 8) /
;  1032     2066  2        (SYM$S_DATA * 8);
;  1033     2067  2
;  1034     2068  2
;  1035     2069  2    ! If the computed offset is not within the allocated vector, the architectural
;  1036     2070  2    ! maximum number of jobs has been reached; return "no space".
;  1037     2071  2    !
;  1038     2072  2    IF .BLOCK_NUMBER GEQU SQH$S_ENTRY_BITMAP_VECTOR/4
;  1039     2073  2    THEN
;  1040     2074  3        BEGIN
;  1041     2075  3        RELEASE_RECORD(SQH$K_RECNO);
;  1042     2076  3        RETURN JBC$_NOQUESPACE;
;  1043     2077  2        END;
;  1044     2078  2
;  1045     2079  2
;  1046     2080  2    ! Allocate and initialize the new record.
;  1047     2081  2    !
;  1048     2082  2    STATUS = ALLOCATE_RECORD( ; SEB_N, SEB);
;  1049     2083  2    IF NOT .STATUS
;  1050     2084  2    THEN
;  1051     2085  3        BEGIN
;  1052     2086  3        RELEASE_RECORD(SQH$K_RECNO);
;  1053     2087  3        RETURN JBC$_NOQUESPACE;
;  1054     2088  2        END;
;  1055     2089  2    VECTOR[SQH[SQH$L_ENTRY_BITMAP_VECTOR], .BLOCK_NUMBER] = .SEB_N;
;  1056     2090  2    SEB[SYM$B_TYPE] = SYM$K_ENTRY_BITMAP;
;  1057     2091  2    BITVECTOR[SEB[SYM$T_DATA], 0] = TRUE;
;  1058     2092  2    REWRITE_RECORD(.SEB_N);
;  1059     2093  2
;  1060     2094  2
;  1061     2095  2    ! Update data structures in the queue header to reflect the new block.
;  1062     2096  2    !
;  1063     2097  2    .P_ENTRY_NUMBER = .SQH[SQH$L_HIGHEST_ENTRY_NUMBER] + 1;
;  1064     2098  2    SQH[SQH$L_NEXT_ENTRY_NUMBER] = .SQH[SQH$L_HIGHEST_ENTRY_NUMBER] + 2;
```

```
;  1065     2099  2 SQH[SQH$L_HIGHEST_ENTRY_NUMBER] =
;  1066     2100  2     .SQH[SQH$L_HIGHEST_ENTRY_NUMBER] + (SYM$S_DATA * 8);
;  1067     2101  2 REWRITE_RECORD(SQH$K_RECNO);
;  1068     2102  2 SS$_NORMAL
;  1069     2103  1 END;


                                        0FFC 00000          .ENTRY  ALLOCATE_ENTRY_NUMBER, Save R2,R3,R4,R5,R6,-;  1928
                                                                    R7,R8,R9,R10,R11
                                59 00000000G    EF  9E 00002        MOVAB   RELEASE_RECORD, R9
                                5E              08  C2 00009        SUBL2   #8, SP
                                                01  DD 0000C        PUSHL   #1                                      1970
                          00000000G     EF      01  FB 0000E        CALLS   #1, READ_RECORD
                                54              50  D0 00015        MOVL    R0, SQH
                                                5A  D4 00018        CLRL    SEB_N                                   1971
                                55        48    A4  9E 0001A        MOVAB   72(SQH), R5                             1977
                                52              65  D0 0001E        MOVL    (R5), ENTRY_NUMBER
                                58        3C    A4  D0 00021        MOVL    60(SQH), ENTRY_NUMBER_LIMIT             1978
                                57              01  D0 00025        MOVL    #1, I                                   1979
                                58              52  D1 00028 1$:    CMPL    ENTRY_NUMBER, ENTRY_NUMBER_LIMIT        1984
                                                74  1A 0002B        BGTRU   8$
                                5B        FF    A2  9E 0002D        MOVAB   -1(R2), BIT_NUMBER                      1989
                          00000800      8F      5B  D1 00031        CMPL    BIT_NUMBER, #2048                       1990
                                                19  1E 00038        BGEQU   2$
                       5D       0100    C4      5B  E2 0003A        BBSS    BIT_NUMBER, 256(SQH), 6$                1998
                                65        01    A2  9E 00040        MOVAB   1(R2), (R5)                             2001
                                  04    BC      52  D0 00044        MOVL    ENTRY_NUMBER, @P_ENTRY_NUMBER           2002
                                                5A  D5 00048        TSTL    SEB_N                                   2003
                                                4F  13 0004A        BEQL    5$
                                                5A  DD 0004C        PUSHL   SEB_N
                                69              01  FB 0004E        CALLS   #1, RELEASE_RECORD
                                                48  11 00051        BRB     5$                                      2004
                                6E      F800    CB  9E 00053 2$:    MOVAB   -2048(R11), Q                           2015
                                      04        AE  D4 00058        CLRL    Q+4                                     2016
                    5B      53          6E 00000FA0 8F  7B 0005B    EDIV    #4000, Q, BLOCK_NUMBER, BIT_NUMBER      2017
                                14 A443             5A  D1 00064    CMPL    SEB_N, 20(SQH)[BLOCK_NUMBER]            2023
                                                    1A  13 00069    BEQL    4$
                                                    5A  D5 0006B    TSTL    SEB_N                                   2026
                                                    05  13 0006D    BEQL    3$
                                                    5A  DD 0006F    PUSHL   SEB_N
                                69                  01  FB 00071    CALLS   #1, RELEASE_RECORD
                                5A            14 A443   D0 00074 3$:    MOVL    20(SQH)[BLOCK_NUMBER], SEB_N    2027
                                                        5A DD 00079     PUSHL   SEB_N                               2028
                          00000000G     EF              01 FB 0007B     CALLS   #1, READ_RECORD
                                56                      50 D0 00082     MOVL    R0, SEB
                       13       0C    A6              5B E2 00085 4$:    BBSS    BIT_NUMBER, 12(SEB), 6$         2035
                                65        01    A2      9E 0008A        MOVAB   1(R2), (R5)                         2038
                                  04    BC              52 D0 0008E     MOVL    ENTRY_NUMBER, @P_ENTRY_NUMBER       2039
                                                        5A DD 00092     PUSHL   SEB_N                               2040
                          00000000G     EF              01 FB 00094     CALLS   #1, REWRITE_RECORD
                                                        70 11 0009B 5$:    BRB     12$                             2041
                                                        52 D6 0009D 6$:    INCL    ENTRY_NUMBER                    2049
                                                        87 11 0009F 7$:    BRB     1$                              1984
                                52                      01 D0 000A1 8$:    MOVL    #1, ENTRY_NUMBER                2055
```

```
                58              65              01  C3 000A4          SUBL3   #1, (R5), ENTRY_NUMBER_LIMIT                ; 2056
                                F4              57  F4 000A8          SOBGEQ  I, 7$                                       ; 1979
                                                5A  D5 000AB          TSTL    SEB_N                                       ; 2063
                                                05  13 000AD          BEQL    9$
                                                5A  DD 000AF          PUSHL   SEB_N
                                            69  01  FB 000B1          CALLS   #1, RELEASE_RECORD
                50      3C      A4 00000800     8F  C3 000B4 9$:      SUBL3   #2048, 60(SQH), R0                           ; 2065
                53              50 00000FA0     8F  C7 000BD          DIVL3   #4000, R0, BLOCK_NUMBER                     ; 2066
                                08              53  D1 000C5          CMPL    BLOCK_NUMBER, #8                            ; 2072
                                                0D  1E 000C8          BGEQU   10$
            00000000G           EF              00  FB 000CA          CALLS   #0, ALLOCATE_RECORD                         ; 2082
                                56              5B  D0 000D1          MOVL    R11, R6
                                0D              50  E8 000D4          BLBS    STATUS, 11$                                 ; 2083
                                                01  DD 000D7 10$:     PUSHL   #1                                          ; 2086
                                            69  01  FB 000D9          CALLS   #1, RELEASE_RECORD
                                50 00048030     8F  D0 000DC          MOVL    #294960, R0                                 ; 2087
                                                    04 000E3          RET
                        14 A443                 5A  D0 000E4 11$:     MOVL    SEB_N, 20(SQH)[BLOCK_NUMBER]                ; 2089
                        04  A6                  0A  90 000E9          MOVB    #10, 4(SEB)                                 ; 2090
                        0C  A6                  01  88 000ED          BISB2   #1, 12(SEB)                                 ; 2091
                                                5A  DD 000F1          PUSHL   SEB_N                                       ; 2092
            00000000G           EF              01  FB 000F3          CALLS   #1, REWRITE_RECORD
        04  BC      3C          A4              01  C1 000FA          ADDL3   #1, 60(SQH), aP_ENTRY_NUMBER                ; 2097
        65      3C              A4              02  C1 00100          ADDL3   #2, 60(SQH), (R5)                           ; 2098
                3C              A4 00000FA0     8F  C0 00105          ADDL2   #4000, 60(SQH)                              ; 2100
                                                01  DD 0010D 12$:     PUSHL   #1                                          ; 2101
            00000000G           EF              01  FB 0010F          CALLS   #1, REWRITE_RECORD
                                50              01  D0 00116          MOVL    #1, R0                                      ; 2103
                                                    04 00119          RET                                                 ;

; Routine Size:  282 bytes,    Routine Base:  CODE + 0568
```

```
;1071     2104  1 GLOBAL ROUTINE DEALLOCATE_ENTRY_NUMBER(ENTRY_NUMBER): NOVALUE=
;1072     2105  1
;1073     2106  1 !++
;1074     2107  1 !
;1075     2108  1 ! FUNCTIONAL DESCRIPTION:
;1076     2109  1 !       This routine deallocates a job entry number.
;1077     2110  1 !
;1078     2111  1 ! INPUT PARAMETERS:
;1079     2112  1 !       ENTRY_NUMBER    - Entry number to be deallocated.
;1080     2113  1 !
;1081     2114  1 ! IMPLICIT INPUTS:
;1082     2115  1 !       NONE
;1083     2116  1 !
;1084     2117  1 ! OUTPUT PARAMETERS:
;1085     2118  1 !       NONE
;1086     2119  1 !
;1087     2120  1 ! IMPLICIT OUTPUTS:
;1088     2121  1 !       NONE
;1089     2122  1 !
;1090     2123  1 ! ROUTINE VALUE:
;1091     2124  1 !       NONE
;1092     2125  1 !
;1093     2126  1 ! SIDE EFFECTS:
;1094     2127  1 !       NONE
;1095     2128  1 !
;1096     2129  1 !--
;1097     2130  1
;1098     2131  2 BEGIN
;1099     2132  2 LOCAL
;1100     2133  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;1101     2134  2         BIT_NUMBER;                     ! Bit number within record
;1102     2135  2
;1103     2136  2
;1104     2137  2 ! Read the queue header.
;1105     2138  2 !
;1106     2139  2 SQH = READ_RECORD(SQH$K_RECNO);
;1107     2140  2
;1108     2141  2
;1109     2142  2 ! Ensure that the entry number is in range.
;1110     2143  2 !
;1111     2144  2 IF .ENTRY_NUMBER EQLU 0
;1112     2145  2 OR .ENTRY_NUMBER GTRU .SQH[SQH$L_HIGHEST_ENTRY_NUMBER]
;1113     2146  2 THEN
;1114     2147  2     RETURN;
;1115     2148  2
;1116     2149  2
;1117     2150  2 ! Determine if the bit is in the queue header or in an extension record,
;1118     2151  2 ! and process accordingly.
;1119     2152  2 !
;1120     2153  2 BIT_NUMBER = .ENTRY_NUMBER - 1;
;1121     2154  2 IF .BIT_NUMBER LSSU SQH$S_ENTRY_BITMAP * 8
;1122     2155  2 THEN
;1123     2156  3     BEGIN
;1124     2157  3     BITVECTOR[SQH[SQH$B_ENTRY_BITMAP], .BIT_NUMBER] = FALSE;
;1125     2158  3     REWRITE_RECORD(SQH$K_RECNO);
;1126     2159  3     END
;1127     2160  2 ELSE
```

```
;  1128          2161  3     BEGIN
;  1129          2162  3     LOCAL
;  1130          2163  3         BLOCK_NUMBER,                   ! Index to extension block
;  1131          2164  3         Q:              VECTOR[2],      ! Temporary for EDIV
;  1132          2165  3         SEB_N,                          ! Record number of extension bitmap
;  1133          2166  3         SEB:            REF BBLOCK;     ! Pointer to extension bitmap
;  1134          2167  3
;  1135          2168  3
;  1136          2169  3     Q[0] = .BIT_NUMBER - SQH$S_ENTRY_BITMAP * 8;
;  1137          2170  3     Q[1] = 0;
;  1138          2171  3     EDIV(%REF(SYM$S_DATA * 8), Q, BLOCK_NUMBER, BIT_NUMBER);
;  1139          2172  3     IF .BLOCK_NUMBER LSSU SQH$S_ENTRY_BITMAP_VECTOR74
;  1140          2173  3     THEN
;  1141          2174  4         BEGIN
;  1142          2175  4         SEB_N = .VECTOR[SQH[SQH$L_ENTRY_BITMAP_VECTOR], .BLOCK_NUMBER];
;  1143          2176  4         IF .SEB_N NEQ 0
;  1144          2177  4         THEN
;  1145          2178  5             BEGIN
;  1146          2179  5             SEB = READ_RECORD(.SEB_N);
;  1147          2180  5             BITVECTOR[SEB[SYM$T_DATA], .BIT_NUMBER] = FALSE;
;  1148          2181  5             REWRITE_RECORD(.SEB_N);
;  1149          2182  4             END;
;  1150          2183  3         END;
;  1151          2184  3     RELEASE_RECORD(SQH$K_RECNO);
;  1152          2185  2     END;
;  1153          2186  1 END;


                                    003C 00000         .ENTRY  DEALLOCATE_ENTRY_NUMBER, Save R2,R3,R4,R5    ; 2104
                              55 00000000G   EF  9E 00002  MOVAB   READ_RECORD, R5                           ;
                              54 00000000G   EF  9E 00009  MOVAB   REWRITE_RECORD, R4                        ;
                              5E             08  C2 00010  SUBL2   #8, SP                                   ;
                                             01  DD 00013  PUSHL   #1                                       ; 2139
                              65             01  FB 00015  CALLS   #1, READ_RECORD                          ;
                              53             50  D0 00018  MOVL    R0, SQH                                  ;
                              52       04    AC  D0 0001B  MOVL    ENTRY_NUMBER, R2                         ; 2144
                                             52  13 0001F  BEQL    5$                                       ;
                        3C    A3             52  D1 00021  CMPL    R2, 60(SQH)                              ; 2145
                                             4C  1A 00025  BGTRU   5$                                       ;
                                             52  D7 00027  DECL    BIT_NUMBER                               ; 2153
                  00000800    8F             52  D1 00029  CMPL    BIT_NUMBER, #2048                        ; 2154
                                             0C  1E 00030  BGEQU   2$                                       ;
              00      0100    C3             52  E5 00032  BBCC    BIT_NUMBER, 256(SQH), 1$                 ; 2157
                              64             01  FB 0003A  CALLS   #1, REWRITE_RECORD                        ; 2158
                                             04  0003D  RET                                                 ; 2154
                              6E      F800   C2  9E 0003E  2$:     MOVAB   -2048(R2), Q                     ; 2169
                                       04    AE  D4 00043  CLRL    Q+4                                      ; 2170
52                50          6E 00000FA0   8F  7B 00046  EDIV    #4000, Q, BLOCK_NUMBER, BIT_NUMBER       ; 2171
                              08             50  D1 0004F  CMPL    BLOCK_NUMBER, #8                         ; 2172
                                             16  1E 00052  BGEQU   4$                                       ;
                              53       14 A340   D0 00054  MOVL    20(SQH)[BLOCK_NUMBER], SEB_N             ; 2175
                                             0F  13 00059  BEQL    4$                                       ; 2176
                                             53  DD 0005B  PUSHL   SEB_N                                    ; 2179
```

```
                                65              01  FB 0005D              CALLS   #1, READ_RECORD                          ;
                          00      0C    A0      52  E5 00060              BBCC    BIT_NUMBER, 12(SEB), 3$                 ; 2180
                                        DD      53  DD 00065 3$:          PUSHL   SEB_N                                   ; 2181
                                64              01  FB 00067              CALLS   #1, REWRITE_RECORD                      ;
                                        DD      01  DD 0006A 4$:          PUSHL   #1                                      ; 2184
                              00000000G EF      01  FB 0006C              CALLS   #1, RELEASE_RECORD                      ;
                                                    04 00073 5$:          RET                                             ; 2186

; Routine Size:  116 bytes,    Routine Base:  CODE + 0682
```

```
;  1155          2187  1 GLOBAL ROUTINE JOB_STATUS_MESSAGE(RESULT,MSG_BUFFER,SMQ,SJH,ESMQ)=
;  1156          2188  1
;  1157          2189  1 !++
;  1158          2190  1 !
;  1159          2191  1 ! FUNCTIONAL DESCRIPTION:
;  1160          2192  1 !       This routine formats a job status message.
;  1161          2193  1 !
;  1162          2194  1 ! INPUT PARAMETERS:
;  1163          2195  1 !       RESULT          - Result of the enqueue.
;  1164          2196  1 !       MSG_BUFFER      - Pointer to message buffer.
;  1165          2197  1 !       SMQ             - Pointer to SMQ.
;  1166          2198  1 !       SJH             - Pointer to SJH.
;  1167          2199  1 !       ESMQ            - Pointer to executor SMQ, if job is executing.
;  1168          2200  1 !
;  1169          2201  1 ! IMPLICIT INPUTS:
;  1170          2202  1 !       NONE
;  1171          2203  1 !
;  1172          2204  1 ! OUTPUT PARAMETERS:
;  1173          2205  1 !       NONE
;  1174          2206  1 !
;  1175          2207  1 ! IMPLICIT OUTPUTS:
;  1176          2208  1 !       NONE
;  1177          2209  1 !
;  1178          2210  1 ! ROUTINE VALUE:
;  1179          2211  1 !       Message length.
;  1180          2212  1 !
;  1181          2213  1 ! SIDE EFFECTS:
;  1182          2214  1 !       NONE
;  1183          2215  1 !
;  1184          2216  1 !--
;  1185          2217  1
;  1186          2218  2 BEGIN
;  1187          2219  2 MAP
;  1188          2220  2         MSG_BUFFER:     REF VECTOR[,BYTE],      ! Pointer to message buffer
;  1189          2221  2         SJH:            REF BBLOCK,             ! Pointer to SJH
;  1190          2222  2         SMQ:            REF BBLOCK,             ! Pointer to SMQ
;  1191          2223  2         ESMQ:           REF BBLOCK;             ! Pointer to SMQ
;  1192          2224  2 LOCAL
;  1193          2225  2         GET_DESC:       VECTOR[2],              ! Descriptor for $GETMSG buffer
;  1194          2226  2         MSG_DESC:       VECTOR[2],              ! Descriptor for message buffer
;  1195          2227  2         LENGTH:         WORD,                   ! Length of message
;  1196          2228  2         PRMLST:         VECTOR[4],              ! $FAOL parameters
;  1197          2229  2         GET_BUFFER:     VECTOR[80,BYTE];        ! $GETMSG buffer
;  1198          2230  2 OWN
;  1199          2231  2         MESSAGES:       VECTOR[5] PSECT(CODE) PRESET(
;  1200          2232  2             [ENQ_K_CURRENT] =   JBC$_NFY_CURRENT,
;  1201          2233  2             [ENQ_K_HOLD] =      JBC$_NFY_HOLD,
;  1202          2234  2             [ENQ_K_PENDING] =   JBC$_NFY_PENDING,
;  1203          2235  2             [ENQ_K_TIMER] =     JBC$_NFY_TIMER,
;  1204          2236  2             [ENQ_K_COMPLETE] =  JBC$_NFY_COMPLETE);
;  1205          2237  2 LITERAL
;  1206          2238  2         MSG_LENGTH=     160;
;  1207          2239  2
;  1208          2240  2
;  1209          2241  2 ! Get the message to be formatted.
;  1210          2242  2 !
;  1211          2243  2 GET_DESC[0] = %ALLOCATION(GET_BUFFER);
```

```
;  1212          2244  2 GET_DESC[1] = GET_BUFFER;
;  1213        P 2245  2 IF NOT $GETMSG(
;  1214        P 2246  2     MSGID=.MESSAGES[.RESULT],
;  1215        P 2247  2     MSGLEN=GET_DESC,
;  1216        P 2248  2     BUFADR=GET_DESC,
;  1217          2249  3     FLAGS=%B'0001')
;  1218          2250  2 THEN
;  1219          2251  2     RETURN 0;
;  1220          2252  2
;  1221          2253  2
;  1222          2254  2 ! Set up the proper FAO parameters.
;  1223          2255  2 !
;  1224          2256  2 PRMLST[0] = SJH[SJH$T_NAME];
;  1225          2257  2 PRMLST[1] = SMQ[SMQ$T_NAME];
;  1226          2258  2 PRMLST[2] = .SJH[SYM$L_ENTRY_NUMBER];
;  1227          2259  2 IF .RESULT EQL ENQ_K_CURRENT THEN PRMLST[3] = ESMQ[SMQ$T_NAME];
;  1228          2260  2 IF .RESULT EQL ENQ_K_TIMER THEN PRMLST[3]= SJH[SJH$Q_AFTER_TIME];
;  1229          2261  2
;  1230          2262  2
;  1231          2263  2 ! Format the message.
;  1232          2264  2 !
;  1233          2265  2 MSG_DESC[0] = MSG_LENGTH;
;  1234          2266  2 MSG_DESC[1] = .MSG_BUFFER;
;  1235        P 2267  2 $FAOL(
;  1236        P 2268  2     CTRSTR=GET_DESC,
;  1237        P 2269  2     OUTLEN=MSG_DESC,
;  1238        P 2270  2     OUTBUF=MSG_DESC,
;  1239          2271  2     PRMLST=PRMLST);
;  1240          2272  2
;  1241          2273  2
;  1242          2274  2 IF .RESULT EQL ENQ_K_COMPLETE
;  1243          2275  2 AND .SJH[SJH$L_CONDITION_1] NEQ 0 AND NOT .SJH[SJH$L_CONDITION_1]
;  1244          2276  2 THEN
;  1245          2277  3     BEGIN
;  1246          2278  3
;  1247          2279  3     ! Append CR-LF to the buffer, and update the descriptor to describe the
;  1248          2280  3     ! remaining portion of the buffer.
;  1249          2281  3     !
;  1250          2282  3     MSG_DESC[1] = .MSG_DESC[1] + .MSG_DESC[0];
;  1251          2283  3     (.MSG_DESC[1])<0,16> = %CHAR(%O'015', %O'012');
;  1252          2284  3     MSG_DESC[1] = .MSG_DESC[1] + 2;
;  1253          2285  3     MSG_DESC[0] = MSG_LENGTH - .MSG_DESC[0] - 2;
;  1254          2286  3
;  1255          2287  3
;  1256          2288  3     ! Get the message corresponding to the completion status.
;  1257          2289  3     !
;  1258        P 2290  3     $GETMSG(
;  1259        P 2291  3         MSGID=.SJH[SJH$L_CONDITION_1],
;  1260        P 2292  3         MSGLEN=LENGTH,
;  1261        P 2293  3         BUFADR=MSG_DESC,
;  1262          2294  3         FLAGS=%B'1111');
;  1263          2295  3
;  1264          2296  3
;  1265          2297  3     ! Update the descriptor to describe the entire message.
;  1266          2298  3     !
;  1267          2299  3     MSG_DESC[0] = MSG_LENGTH - .MSG_DESC[0] + .LENGTH;
;  1268          2300  2     END;
```

```
;  1269          2301  2
;  1270          2302  2
;  1271          2303  2 ,MSG_DESC[0]
;  1272          2304  1 END;


                                                006F6          .BLKB   2
        00048480  00048A0  00048498  00048490  00048488  006F8 MESSAGES:
                                                               .LONG   296072, 296080, 296088, 296096, 296064      ;

                                                               .EXTRN  SYS$GETMSG, SYS$FAOL

                                                001C 00000     .ENTRY  JOB_STATUS_MESSAGE, Save R2,R3,R4           ; 2187
                          54 00000000G   00     9E 00002     MOVAB   SYS$GETMSG, R4
                          5E      8C     AE     9E 00009     MOVAB   -116(SP), SP
                      6C  AE      50     8F     9A 0000D     MOVZBL  #80, GET_DESC                               ; 2243
                      70  AE      04     AE     9E 00012     MOVAB   GET_BUFFER, GET_DESC+4                      ; 2244
                          7E              01     7D 00017     MOVQ    #1, -(SP)                                   ; 2249
                                  74     AE     9F 0001A     PUSHAB  GET_DESC
                                  78     AE     9F 0001D     PUSHAB  GET_DESC
                          53      04     AC     D0 00020     MOVL    RESULT, R3
                                  C4 AF43        DD 00024     PUSHL   MESSAGES[R3]
                          64              05     FB 00028     CALLS   #5, SYS$GETMSG
                          03              50     E8 0002B     BLBS    R0, 1$
                                          50     D4 0002E     CLRL    R0                                          ; 2251
                                          04     00030        RET
                          52      10      AC     D0 00031 1$: MOVL    SJH, R2                                     ; 2256
                      54  AE    0108      C2     9E 00035     MOVAB   264(R2), PRMLST
                58    AE      0C  AC 00000080  8F     C1 0003B     ADDL3   #176, SMQ, PRMLST+4                     ; 2257
                      5C  AE      08     A2     D0 00045     MOVL    8(R2), PRMLST+8                             ; 2258
                                          53     D5 0004A     TSTL    R3                                          ; 2259
                                          0A     12 0004C     BNEQ    2$
                60    AE        14  AC 00000080  8F     C1 0004E     ADDL3   #176, ESMQ, PRMLST+12
                          03              53     D1 00058 2$: CMPL    R3, #3                                      ; 2260
                                          06     12 0005B     BNEQ    3$
                      60  AE      0098      C2     9E 0005D     MOVAB   152(R2), PRMLST+12
                      64  AE        A0      8F     9A 00063 3$: MOVZBL  #160, MSG_DESC                          ; 2265
                      68  AE        08      AC     D0 00068     MOVL    MSG_BUFFER, MSG_DESC+4                      ; 2266
                                    54     AE     9F 0006D     PUSHAB  PRMLST                                      ; 2271
                                    68     AE     9F 00070     PUSHAB  MSG_DESC
                                    6C     AE     9F 00073     PUSHAB  MSG_DESC
                                    78     AE     9F 00076     PUSHAB  GET_DESC
                    00000000G   00                  04     FB 00079     CALLS   #4, SYS$FAOL
                          04                        53     D1 00080     CMPL    R3, #4                              ; 2274
                                                    41     12 00083     BNEQ    4$
                                  00DC      C2     D5 00085     TSTL    220(R2)                                     ; 2275
                                                    3B     13 00089     BEQL    4$
                          36      00DC      C2     E8 0008B     BLBS    220(R2), 4$
                      68  AE        64      AE     C0 00090     ADDL2   MSG_DESC, MSG_DESC+4                      ; 2282
                      68  BE      0A0D      8F     B0 00095     MOVW    #2573, @MSG_DESC+4                          ; 2283
                      68  AE                02     C0 0009B     ADDL2   #2, MSG_DESC+4                              ; 2284
64    AE  0000009E      8F        64      AE     C3 0009F     SUBL3   MSG_DESC, #158, MSG_DESC                    ; 2285
                          7E              0F     7D 000A9     MOVQ    #15, -(SP)                                  ; 2294
                                    6C     AE     9F 000AC     PUSHAB  MSG_DESC
                                    0C     AE     9F 000AF     PUSHAB  LENGTH
                                  00DC      C2     DD 000B2     PUSHL   220(R2)
```

```
                64              05  FB 000B6          CALLS   #5, SYS$GETMSG                          ;
                50              6E  3C 000B9          MOVZWL  LENGTH, R0                             ; 2299
                50        64    AE  C2 000BC          SUBL2   MSG_DESC, R0                           ;
          64    AE      00A0  C0  9E 000C0          MOVAB   160(R0), MSG_DESC                      ;
                50        64    AE  D0 000C6 4$:      MOVL    MSG_DESC, R0                           ; 2303
                                    04 000CA          RET                                            ; 2304

; Routine Size:  203 bytes,    Routine Base:  CODE + 070C
```

```
;  1274          2305  1 ROUTINE NOTIFY_USER(RESULT,SMQ,SJH,ESMQ): NOVALUE=
;  1275          2306  1
;  1276          2307  1 !++
;  1277          2308  1 !
;  1278          2309  1 ! FUNCTIONAL DESCRIPTION:
;  1279          2310  1 !       This routine notifies a user via broadcast of the status of a job.
;  1280          2311  1 !
;  1281          2312  1 ! INPUT PARAMETERS:
;  1282          2313  1 !       RESULT          - Result of the enqueue.
;  1283          2314  1 !       SMQ             - Pointer to SMQ.
;  1284          2315  1 !       SJH             - Pointer to SJH.
;  1285          2316  1 !       ESMQ            - Pointer to executor SMQ, if job is executing.
;  1286          2317  1 !
;  1287          2318  1 ! IMPLICIT INPUTS:
;  1288          2319  1 !       NONE
;  1289          2320  1 !
;  1290          2321  1 ! OUTPUT PARAMETERS:
;  1291          2322  1 !       NONE
;  1292          2323  1 !
;  1293          2324  1 ! IMPLICIT OUTPUTS:
;  1294          2325  1 !       NONE
;  1295          2326  1 !
;  1296          2327  1 ! ROUTINE VALUE:
;  1297          2328  1 !       NONE
;  1298          2329  1 !
;  1299          2330  1 ! SIDE EFFECTS:
;  1300          2331  1 !       Messages broadcast to terminals.
;  1301          2332  1 !
;  1302          2333  1 !--
;  1303          2334  1
;  1304          2335  2 BEGIN
;  1305          2336  2 MAP
;  1306          2337  2         SJH:            REF BBLOCK,             ! Pointer to SJH
;  1307          2338  2         SMQ:            REF BBLOCK,             ! Pointer to SMQ
;  1308          2339  2         ESMQ:           REF BBLOCK;             ! Pointer to SMQ
;  1309          2340  2 LOCAL
;  1310          2341  2         LENGTH,                                 ! Length of message
;  1311          2342  2         MSG_BUFFER:     VECTOR[SRQ$S_BRDCST_TEXT,BYTE]; ! Message buffer
;  1312          2343  2
;  1313          2344  2
;  1314          2345  2 ! Fill in a CR-LF and two bells ahead of the message, and get the formatted
;  1315          2346  2 ! message.
;  1316          2347  2 !
;  1317          2348  2 MSG_BUFFER = %CHAR(%O'015', %O'012', %O'007', %O'007');
;  1318          2349  2 LENGTH = 4 + JOB_STATUS_MESSAGE(.RESULT, MSG_BUFFER + 4, .SMQ, .SJH, .ESMQ);
;  1319          2350  2
;  1320          2351  2
;  1321          2352  2 ! Issue the broadcast message.
;  1322          2353  2 !
;  1323          2354  2 BROADCAST_MESSAGE(
;  1324          2355  2     SJH[SJH$T_SYSID],
;  1325          2356  2     SJH[SJH$T_USERNAME],
;  1326          2357  2     .LENGTH, MSG_BUFFER);
;  1327          2358  1 END;
```

```
                                        0000 00000 NOTIFY_USER:
                                                           .WORD   Save nothing                                    ; 2305
                                SE      FE44    CE  9E 00002       MOVAB   -444(SP), SP                            ;
                             07070A0D           8F  DD 00007       PUSHL   #117901837                              ; 2348
                                7E        0C    AC  7D 0000D       MOVQ    SJH, -(SP)                              ; 2349
                                          08    AC  DD 00011       PUSHL   SMQ                                     ;
                                          10    AE  9F 00014       PUSHAB  MSG_BUFFER+4                            ;
                                          04    AC  DD 00017       PUSHL   RESULT                                  ;
                          FF16    CF              05  FB 0001A       CALLS   #5, JOB_STATUS_MESSAGE                  ;
                                  50              04  C0 0001F       ADDL2   #4, LENGTH                              ;
                                        4001      8F  BB 00022       PUSHR   #^M<R0,SP>                              ; 2357
                       7E         0C    AC 00000148 8F  C1 00026       ADDL3   #328, SJH, -(SP)                        ; 2356
                       7E         0C    AC 0000016C 8F  C1 0002F       ADDL3   #364, SJH, -(SP)                        ; 2355
                          00000000G      EF          04  FB 00038       CALLS   #4, BROADCAST_MESSAGE                   ; 2356
                                                     04  0003F       RET                                             ; 2358

; Routine Size:  64 bytes,    Routine Base:  CODE + 07D7
```

```
;  1329     2359  1 GLOBAL ROUTINE COMPLETE_JOB(SJH_N,SJH,SMQ,ACM,STS): NOVALUE=
;  1330     2360  1
;  1331     2361  1 !++
;  1332     2362  1 !
;  1333     2363  1 ! FUNCTIONAL DESCRIPTION:
;  1334     2364  1 !       This routine completes a job.
;  1335     2365  1 !
;  1336     2366  1 ! INPUT PARAMETERS:
;  1337     2367  1 !       SJH_N           - Record number of SJH.
;  1338     2368  1 !       SJH             - Pointer to SJH.
;  1339     2369  1 !       SMQ             - Pointer to SMQ.
;  1340     2370  1 !       ACM             - Pointer to ACM or 0.
;  1341     2371  1 !       STS             - (Optional) Forced completion status.
;  1342     2372  1 !
;  1343     2373  1 ! IMPLICIT INPUTS:
;  1344     2374  1 !       NONE
;  1345     2375  1 !
;  1346     2376  1 ! OUTPUT PARAMETERS:
;  1347     2377  1 !       NONE
;  1348     2378  1 !
;  1349     2379  1 ! IMPLICIT OUTPUTS:
;  1350     2380  1 !       NONE
;  1351     2381  1 !
;  1352     2382  1 ! ROUTINE VALUE:
;  1353     2383  1 !       NONE
;  1354     2384  1 !
;  1355     2385  1 ! SIDE EFFECTS:
;  1356     2386  1 !       NONE
;  1357     2387  1 !
;  1358     2388  1 !--
;  1359     2389  1
;  1360     2390  2 BEGIN
;  1361     2391  2 MAP
;  1362     2392  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  1363     2393  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  1364     2394  2         ACM:            REF BBLOCK;     ! Pointer to ACM
;  1365     2395  2 BUILTIN
;  1366     2396  2         ACTUALCOUNT;
;  1367     2397  2
;  1368     2398  2
;  1369     2399  2 ! First, check to see that the SJH is valid by verifying the QUEUE_LINK is
;  1370     2400  2 ! non-zero.  Invalid SJHs can only occur in a corrupted file.  An invalid
;  1371     2401  2 ! SJH is generally a record that has been deallocated to the free list but
;  1372     2402  2 ! still may appear in the SMQ's current list.
;  1373     2403  2 !
;  1374     2404  2 IF .SJH[SJH$L_QUEUE_LINK] EQL 0 THEN RETURN;
;  1375     2405  2
;  1376     2406  2 ! Propagate the process termination status to the SJH record.
;  1377     2407  2 !
;  1378     2408  2 IF .SJH[SJH$L_CONDITION_1] EQL 0 AND .ACM NEQ 0
;  1379     2409  2 THEN
;  1380     2410  3     BEGIN
;  1381     2411  3     SJH[SJH$L_CONDITION_1] = .ACM[ACM$L_FINALSTS];
;  1382     2412  3     SJH[SJH$L_CONDITION_2] = 0;
;  1383     2413  3     SJH[SJH$L_CONDITION_3] = 0;
;  1384     2414  2     END;
;  1385     2415  2
```

```
;  1386     2416  2
;  1387     2417  2 ! Propagate the forced abort, requeue, or delete status, if specified.
;  1388     2418  2 !
;  1389     2419  2 IF .SJH[SJH$V_DELETED]
;  1390     2420  2 THEN
;  1391     2421  2     BEGIN
;  1392     2422  2     SJH[SJH$L_CONDITION_1] = JBC$_JOBDELETE OR STS$K_ERROR;
;  1393     2423  2     SJH[SJH$L_CONDITION_2] = 0;
;  1394     2424  2     SJH[SJH$L_CONDITION_3] = 0;
;  1395     2425  2     END
;  1396     2426  2
;  1397     2427  2 ELSE IF .SJH[SJH$V_ABORTED]
;  1398     2428  2 THEN
;  1399     2429  2     BEGIN
;  1400     2430  2     IF .SJH[SJH$V_REQUEUE]
;  1401     2431  3         THEN SJH[SJH$L_CONDITION_1] = JBC$_JOBREQUEUE OR STS$K_ERROR
;  1402     2432  3         ELSE SJH[SJH$L_CONDITION_1] = JBC$_JOBABORT OR STS$K_ERROR;
;  1403     2433  2     SJH[SJH$L_CONDITION_2] = 0;
;  1404     2434  2     SJH[SJH$L_CONDITION_3] = 0;
;  1405     2435  2     END;
;  1406     2436  2
;  1407     2437  2
;  1408     2438  2 ! Propagate the forced completion status, if specified.
;  1409     2439  2 !
;  1410     2440  2 IF ACTUALCOUNT() GEQU 5
;  1411     2441  2 THEN
;  1412     2442  2     BEGIN
;  1413     2443  2     SJH[SJH$L_CONDITION_1] = .STS;
;  1414     2444  2     SJH[SJH$L_CONDITION_2] = 0;
;  1415     2445  2     SJH[SJH$L_CONDITION_3] = 0;
;  1416     2446  2     END;
;  1417     2447  2
;  1418     2448  2
;  1419     2449  2 ! Write an accounting record for the job except if it was not executing
;  1420     2450  2 ! at the time of a system failure, or it has been retained.
;  1421     2451  2 !
;  1422     2452  3 IF (NOT .SJH[SJH$V_SYSTEM_FAILURE] OR .SJH[SJH$V_EXECUTING])
;  1423     2453  2 AND NOT .SJH[SJH$V_RETAINED]
;  1424     2454  2 THEN
;  1425     2455  2     WRITE_ACCOUNTING_RECORD(.SJH, .SMQ, .ACM);
;  1426     2456  2
;  1427     2457  2
;  1428     2458  2 ! Delete jobs from the completed queue without going through NOTIFY and
;  1429     2459  2 ! SYNCHRONIZE processing.
;  1430     2460  2 !
;  1431     2461  2 IF .SJH[SJH$V_DELETED]
;  1432     2462  2 AND .SJH[SJH$V_RETAINED]
;  1433     2463  2 THEN
;  1434     2464  2     DELETE_SJH_RECORD(.SJH_N, .SJH)
;  1435     2465  2
;  1436     2466  2
;  1437     2467  2 ! Requeue the job if required.
;  1438     2468  2 !
;  1439     2469  2 ELSE IF .SJH[SJH$V_SYSTEM_FAILURE]
;  1440     2470  3     AND (NOT .SJH[SJH$V_EXECUTING] OR .SJH[SJH$V_RESTART])
;  1441     2471  2 OR .SJH[SJH$V_REQUEUE]
;  1442     2472  2 OR .SJH[SJH$V_RETAINED]
```

```
;  1443     2473  2 THEN
;  1444     2474  2     BEGIN
;  1445     2475  2     SJH[SJH$L_CURRENT_FILE_LINK] = 0;
;  1446     2476  2     ENQUEUE_JOB(.SJH_N, .SJH);
;  1447     2477  2     REWRITE_RECORD(.SJH_N);
;  1448     2478  2     END
;  1449     2479  2
;  1450     2480  2
;  1451     2481  2 ! Complete the job with NOTIFY and SYNCHRONIZE processing, and then delete or
;  1452     2482  2 ! retain the job according to the /RETAIN specification.
;  1453     2483  2 !
;  1454     2484  2 ELSE
;  1455     2485  2     BEGIN
;  1456     2486  2     LOCAL
;  1457     2487  2         QSMQ_N,                     ! Record number of job's SMQ
;  1458     2488  2         QSMQ:           REF BBLOCK; ! Pointer to job's SMQ
;  1459     2489  2
;  1460     2490  2
;  1461     2491  2     ! If the /NOTIFY qualifier was given, send the completion notification to
;  1462     2492  2     ! the user.
;  1463     2493  2     !
;  1464     2494  2     IF .SJH[SJH$V_NOTIFY]
;  1465     2495  2     THEN
;  1466     2496  2         NOTIFY_USER(ENQ_K_COMPLETE, .SMQ, .SJH, 0);
;  1467     2497  2
;  1468     2498  2
;  1469     2499  2     ! If there are SYNCHRONIZE commands pending for this job, send the response
;  1470     2500  2     ! messages.
;  1471     2501  2     !
;  1472     2502  2     IF .SJH[SJH$V_SYNCHRONIZE]
;  1473     2503  2     THEN
;  1474     2504  2         SCAN_INCOMPLETE_SERVICES(
;  1475     2505  2             ISRV_K_SYNCHRONIZE,
;  1476     2506  2             .SJH_N,
;  1477     2507  2             .SJH[SJH$L_CONDITION_1]);
;  1478     2508  2
;  1479     2509  2
;  1480     2510  2     ! Read the job's queue record.
;  1481     2511  2     !
;  1482     2512  2     QSMQ = READ_RECORD(QSMQ_N = .SJH[SJH$L_QUEUE_LINK]);
;  1483     2513  2
;  1484     2514  2
;  1485     2515  2     ! If the job is to be retained, do so; otherwise delete it.
;  1486     2516  2     !
;  1487     2517  3     IF NOT .SJH[SJH$V_DELETED]
;  1488     2518  4     AND (.QSMQ[SMQ$V_RETAIN_ALL_JOBS]
;  1489     2519  4         OR (.QSMQ[SMQ$V_RETAIN_ERROR_JOBS] AND NOT .SJH[SJH$L_CONDITION_1]))
;  1490     2520  3     THEN
;  1491     2521  4         BEGIN
;  1492     2522  4         SJH[SJH$L_COMPLETED_BLOCKS] = 0;
;  1493     2523  4         SJH[SJH$L_CURRENT_FILE_CHKPT] = 0;
;  1494     2524  4         SJH[SJH$B_JOB_COPIES_CHKPT] = 0;
;  1495     2525  4         SJH[SJH$B_FILE_COPIES_CHKPT] = 0;
;  1496     2526  4         DEALLOCATE_VARIABLE_DATA(
;  1497     2527  4             SJH$S_CHECKPOINT,
;  1498     2528  4             SJH[SJH$T_CHECKPOINT]);
;  1499     2529  4         SJH[SJH$L_CURRENT_FILE_LINK] = 0;
```

```
;  1500          2530  4          SJH[SJH$V_RETAINED] = TRUE;
;  1501          2531  4          ENQUEUE_JOB(.SJH_N, .SJH);
;  1502          2532  4          REWRITE_RECORD(.SJH_N);
;  1503          2533  4          END
;  1504          2534  3      ELSE
;  1505          2535  3          DELETE_SJH_RECORD(.SJH_N, .SJH);
;  1506          2536  3
;  1507          2537  3
;  1508          2538  3      RELEASE_RECORD(.QSMQ_N);
;  1509          2539  2      END;
;  1510          2540  1  END;


                                    0FFC 00000          .ENTRY  COMPLETE_JOB, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 2359
                                                                R10,R11
                          59 00000000G  EF  9E 00002            MOVAB   REWRITE_RECORD, R9
                          58 00000000G  EF  9E 00009            MOVAB   ENQUEUE_JOB, R8
                          57 00000000G  EF  9E 00010            MOVAB   DELETE_SJH_RECORD, R7
                          52       08   AC  D0 00017            MOVL    SJH, R2                                 ; 2404
                                 0134   C2  D5 0001B            TSTL    308(R2)
                                   01   12  12 0001F            BNEQ    1$
                                        04  00021            RET
                          54       00DC  C2  9E 00022 1$:       MOVAB   220(R2), R4                             ; 2408
                                        64  D5 00027            TSTL    (R4)
                                        11  12 00029            BNEQ    2$
                                     10 AC  D5 0002B            TSTL    ACM
                                        0C  13 0002E            BEQL    2$
                                50      10 AC  D0 00030         MOVL    ACM, R0                                 ; 2411
                                64      4C AO  D0 00034         MOVL    76(R0), (R4)
                                      00E0 C2  7C 00038         CLRQ    224(R2)                                 ; 2412
                                53      10 A2  9E 0003C 2$:     MOVAB   16(R2), R3                              ; 2419
                             09         63 02  E1 00040         BBC     #2, (R3), 3$
                                64 000480D2 8F  D0 00044         MOVL    #295122, (R4)                           ; 2422
                                           17  11 0004B         BRB     5$                                      ; 2423
                                18         63  E9 0004D 3$:     BLBC    (R3), 6$                                ; 2427
                                09      01 A3  E9 00050         BLBC    1(R3), 4$                               ; 2430
                                64 000480E2 8F  D0 00054         MOVL    #295138, (R4)                           ; 2431
                                           07  11 0005B         BRB     5$
                                64 00048082 8F  D0 0005D 4$:     MOVL    #295042, (R4)                           ; 2432
                                      00E0 C2  7C 00064 5$:     CLRQ    224(R2)                                 ; 2433
                                05         6C  91 00068 6$:     CMPB    (AP), #5                                ; 2440
                                           08  1F 0006B         BLSSU   7$
                                64      14 AC  D0 0006D         MOVL    STS, (R4)                               ; 2443
                                      00E0 C2  7C 00071         CLRQ    224(R2)                                 ; 2444
                          04            63  0E  E1 00075 7$:     BBC     #14, (R3), 8$                           ; 2452
                          11            63  03  E1 00079         BBC     #3, (R3), 9$
                          0D            63  0B  E0 0007D 8$:     BBS     #11, (R3), 9$                           ; 2453
                                7E      0C AC  7D 00081         MOVQ    SMQ, -(SP)                              ; 2455
                                           52  DD 00085         PUSHL   R2
                             00000000G  EF     03  FB 00087      CALLS   #3, WRITE_ACCOUNTING_RECORD
                                55      04 AC  D0 0008E 9$:     MOVL    SJH_N, R5                               ; 2464
                          0C            63  02  E1 00092         BBC     #2, (R3), 10$                           ; 2461
                          08            63  0B  E1 00096         BBC     #11, (R3), 10$                          ; 2462
                                           52  DD 0009A         PUSHL   R2                                      ; 2464
```

```
                                  55  DD 0009C          PUSHL   R5                                      ;
                   67             02  FB 0009E          CALLS   #2, DELETE_SJH_RECORD                   ;
                                  04  000A1             RET                                             ;
09                 63             0E  E1 000A2  10$:    BBC     #14, (R3), 11$                          ; 2469
0D                 63             03  E1 000A6          BBC     #3, (R3), 12$                           ; 2470
08        0E       A2             01  E0 000AA          BBS     #1, 14(R2), 12$                         ;
                   04      01     A3  E8 000AF  11$:    BLBS    1(R3), 12$                              ; 2471
11                 63             0B  E1 000B3          BBC     #11, (R3), 13$                          ; 2472
                         00F0     C2  D4 000B7  12$:    CLRL    240(R2)                                 ; 2475
                                  52  DD 000BB          PUSHL   R2                                      ; 2476
                                  55  DD 000BD          PUSHL   R5                                      ;
                   68             02  FB 000BF          CALLS   #2, ENQUEUE_JOB                         ;
                                  55  DD 000C2          PUSHL   R5                                      ; 2477
                   69             01  FB 000C4          CALLS   #1, REWRITE_RECORD                      ;
                                  04  000C7             RET                                             ; 2469
0E        0D       A2             06  E1 000C8  13$:    BBC     #6, 13(R2), 14$                         ; 2494
                                  7E  D4 000CD          CLRL    -(SP)                                   ; 2496
                                  52  DD 000CF          PUSHL   R2                                      ;
                           0C     AC  DD 000D1          PUSHL   SMQ                                     ;
                                  04  DD 000D4          PUSHL   #4                                      ;
          FEE5     CF             04  FB 000D6          CALLS   #4, NOTIFY_USER                         ;
0E                 63             0D  E1 000DB  14$:    BBC     #13, (R3), 15$                          ; 2502
                                  64  DD 000DF          PUSHL   (R4)                                    ; 2507
                           04     AC  DD 000E1          PUSHL   SJH_N                                   ; 2506
                                  01  DD 000E4          PUSHL   #1                                      ; 2504
     00000000G     EF             03  FB 000E6          CALLS   #3, SCAN_INCOMPLETE_SERVICES            ;
                   56    0134     C2  D0 000ED  15$:    MOVL    308(R2), QSMQ_N                         ; 2512
                                  56  DD 000F2          PUSHL   QSMQ_N                                  ;
     00000000G     EF             01  FB 000F4          CALLS   #1, READ_RECORD                         ;
3E                 63             02  E0 000FB          BBS     #2, (R3), 17$                           ; 2517
08        0E       A0             02  E0 000FF          BBS     #2, 14(QSMQ), 16$                       ; 2518
34        0E       A0             03  E1 00104          BBC     #3, 14(QSMQ), 17$                       ; 2519
                   31             64  E8 00109          BLBS    (R4), 17$                               ;
                         00D8     C2  D4 0010C  16$:    CLRL    216(R2)                                 ; 2522
                         00EC     C2  D4 00110          CLRL    236(R2)                                 ; 2523
                         017B     C2  94 00114          CLRB    379(R2)                                 ; 2524
                         0178     C2  94 00118          CLRB    376(R2)                                 ; 2525
                         0180     C2  9F 0011C          PUSHAB  384(R2)                                 ; 2528
                                  20  DD 00120          PUSHL   #32                                     ;
         0000V     CF             02  FB 00122          CALLS   #2, DEALLOCATE_VARIABLE_DATA            ;
                         00F0     C2  D4 00127          CLRL    240(R2)                                 ; 2529
             01    A3             08  88 0012B          BISB2   #8, 1(R3)                               ; 2530
                                  52  DD 0012F          PUSHL   R2                                      ; 2531
                                  55  DD 00131          PUSHL   R5                                      ;
                   68             02  FB 00133          CALLS   #2, ENQUEUE_JOB                         ;
                                  55  DD 00136          PUSHL   R5                                      ; 2532
                   69             01  FB 00138          CALLS   #1, REWRITE_RECORD                      ;
                                  07  11 0013B          BRB     18$                                     ; 2517
                                  52  DD 0013D  17$:    PUSHL   R2                                      ; 2535
                                  55  DD 0013F          PUSHL   R5                                      ;
                   67             02  FB 00141          CALLS   #2, DELETE_SJH_RECORD                   ;
                                  56  DD 00144  18$:    PUSHL   QSMQ_N                                  ; 2538
     00000000G     EF             01  FB 00146          CALLS   #1, RELEASE_RECORD                      ;
                                  04  0014D             RET                                             ; 2540

; Routine Size:  334 bytes,    Routine Base:  CODE + 0817
```

```
;  1512     2541  1 GLOBAL ROUTINE VALIDATE_OBJECT_NAME(LENGTH,ADDRESS,DESC)=
;  1513     2542  1
;  1514     2543  1 !++
;  1515     2544  1 !
;  1516     2545  1 ! FUNCTIONAL DESCRIPTION:
;  1517     2546  1 !       This routine validates a characteristic, form, or queue name.
;  1518     2547  1 !
;  1519     2548  1 ! INPUT PARAMETERS:
;  1520     2549  1 !       LENGTH          - Descriptor for ASCII name.
;  1521     2550  1 !       ADDRESS         -
;  1522     2551  1 !
;  1523     2552  1 ! IMPLICIT INPUTS:
;  1524     2553  1 !       NONE
;  1525     2554  1 !
;  1526     2555  1 ! OUTPUT PARAMETERS:
;  1527     2556  1 !       DESC            - Short descriptor for converted name.
;  1528     2557  1 !
;  1529     2558  1 ! IMPLICIT OUTPUTS:
;  1530     2559  1 !       NONE
;  1531     2560  1 !
;  1532     2561  1 ! ROUTINE VALUE:
;  1533     2562  1 !       True if the parameter is a valid name, false otherwise.
;  1534     2563  1 !
;  1535     2564  1 ! SIDE EFFECTS:
;  1536     2565  1 !       NONE
;  1537     2566  1 !
;  1538     2567  1 !--
;  1539     2568  1
;  1540     2569  2 BEGIN
;  1541     2570  2 MAP
;  1542     2571  2         ADDRESS:        REF VECTOR[,BYTE],      ! Pointer to ASCIC queue name
;  1543     2572  2         DESC:           REF BBLOCK;             ! Pointer to short descriptor
;  1544     2573  2
;  1545     2574  2
;  1546     2575  2 ! Ensure that the length is valid.
;  1547     2576  2 !
;  1548     2577  2 IF .LENGTH EQL 0 OR .LENGTH GTRU 31
;  1549     2578  2 THEN
;  1550     2579  2     RETURN FALSE;
;  1551     2580  2
;  1552     2581  2
;  1553     2582  2 ! Initialize the descriptor.
;  1554     2583  2 !
;  1555     2584  2 DESC[SDSC_W_LENGTH] = .LENGTH;
;  1556     2585  2 DESC[SDSC_A_POINTER] = .ADDRESS;
;  1557     2586  2
;  1558     2587  2
;  1559     2588  2 ! Loop through all characters of the name checking for validity.  Convert
;  1560     2589  2 ! lowercase to uppercase in place, and remove a trailing colon if one exists.
;  1561     2590  2 !
;  1562     2591  2 INCR I FROM 0 TO .LENGTH-1 DO
;  1563     2592  3     BEGIN
;  1564     2593  3     LOCAL
;  1565     2594  3         C:      BYTE;
;  1566     2595  3
;  1567     2596  3     C = .ADDRESS[.I];
;  1568     2597  3     SELECTONE .C OF
```

```
;  1569          2598  3          SET
;  1570          2599               [%C'A' TO %C'Z', %C'0' TO %C'9', %C'$', %C'_']:
;  1571          2600                   0;
;  1572          2601               [%C'a' TO %C'z']:
;  1573          2602                   ADDRESS[.I] = .ADDRESS[.I] - %C'a' + %C'A';
;  1574          2603               [%C':']:
;  1575          2604                   IF .I EQL .LENGTH-1
;  1576          2605                       THEN DESC[SDSC_W_LENGTH] = .DESC[SDSC_W_LENGTH] - 1
;  1577          2606                       ELSE RETURN FALSE;
;  1578          2607               [OTHERWISE]:
;  1579          2608                   RETURN FALSE;
;  1580          2609               TES;
;  1581          2610           END;
;  1582          2611
;  1583          2612
;  1584          2613  2    TRUE
;  1585          2614  1    END;
```

```
                                    000C 00000          .ENTRY  VALIDATE_OBJECT_NAME, Save R2,R3                ; 2541
                          53    04    AC  D0 00002          MOVL    LENGTH, R3                                  ; 2577
                                      61  13 00006          BEQL    5$
                          1F            53  D1 00008          CMPL    R3, #31
                                        5C  1A 0000B          BGTRU   5$
                          52    0C    AC  D0 0000D          MOVL    DESC, R2                                    ; 2584
                          62          53  B0 00011          MOVW    R3, (R2)
                       02 A2    08    AC  D0 00014          MOVL    ADDRESS, 2(R2)                              ; 2585
                          50          01  CE 00019          MNEGL   #1, I                                       ; 2591
                                      4E  11 0001C          BRB     6$
                          51    08 BC40   90 0001E  1$:     MOVB    @ADDRESS[I], C                              ; 2596
                          24          51  91 00023          CMPB    C, #36                                      ; 2599
                                      44  13 00026          BEQL    6$
                          30          51  91 00028          CMPB    C, #48
                                      05  1F 0002B          BLSSU   2$
                          39          51  91 0002D          CMPB    C, #57
                                      3A  1B 00030          BLEQU   6$
                       41 8F          51  91 00032  2$:     CMPB    C, #65
                                      06  1F 00036          BLSSU   3$
                       5A 8F          51  91 00038          CMPB    C, #90
                                      2E  1B 0003C          BLEQU   6$
                       5F 8F          51  91 0003E  3$:     CMPB    C, #95
                                      28  13 00042          BEQL    6$
                       61 8F          51  91 00044          CMPB    C, #97                                      ; 2601
                                      0D  1F 00048          BLSSU   4$
                       7A 8F          51  91 0004A          CMPB    C, #122
                                      07  1A 0004E          BGTRU   4$
                    08 BC40           20  82 00050          SUBB2   #32, @ADDRESS[I]                            ; 2602
                                      15  11 00055          BRB     6$
                          3A          51  91 00057  4$:     CMPB    C, #58                                      ; 2603
                                      0D  12 0005A          BNEQ    5$
                          51    FF    A3  9E 0005C          MOVAB   -1(R3), R1                                  ; 2604
                          51            50  D1 00060          CMPL    I, R1
                                      04  12 00063          BNEQ    5$
                                      62  B7 00065          DECW    (R2)                                        ; 2605
```

```
                         03  11 00067              BRB     6$
                         50  D4 00069 5$:           CLRL    R0                                      ;  2608
                             04 0006B               RET                                             ;
                      AE     50  53  F2 0006C 6$:    AOBLSS  R3, I, 1$                               ;  2591
                             50  01  D0 00070        MOVL    #1, R0                                  ;  2614
                                 04 00073           RET                                             ;

; Routine Size:  116 bytes,    Routine Base:  CODE + 0965
```

```
;  1587          2615  1 GLOBAL ROUTINE FIND_CHARACTERISTIC(DESC;NUMBER): L_OUTPUT_1=
;  1588          2616  1
;  1589          2617  1 !++
;  1590          2618  1 !
;  1591          2619  1 ! FUNCTIONAL DESCRIPTION:
;  1592          2620  1 !       This routine looks up a characteristic name.
;  1593          2621  1 !
;  1594          2622  1 ! INPUT PARAMETERS:
;  1595          2623  1 !       DESC            - Short descriptor for characteristic name.
;  1596          2624  1 !
;  1597          2625  1 ! IMPLICIT INPUTS:
;  1598          2626  1 !       NONE
;  1599          2627  1 !
;  1600          2628  1 ! OUTPUT PARAMETERS:
;  1601          2629  1 !       NUMBER          - Numeric equivalent of the characteristic name.
;  1602          2630  1 !
;  1603          2631  1 ! IMPLICIT OUTPUTS:
;  1604          2632  1 !       NONE
;  1605          2633  1 !
;  1606          2634  1 ! ROUTINE VALUE:
;  1607          2635  1 !       True if the characteristic is defined, false otherwise.
;  1608          2636  1 !
;  1609          2637  1 ! SIDE EFFECTS:
;  1610          2638  1 !       NONE
;  1611          2639  1 !
;  1612          2640  1 !--
;  1613          2641  1
;  1614          2642  2 BEGIN
;  1615          2643  2 MAP
;  1616          2644  2         DESC:           REF BBLOCK;     ! Short descriptor for name
;  1617          2645  2 LOCAL
;  1618          2646  2         SCX:            REF BBLOCK,     ! Pointer to SCX
;  1619          2647  2         SCX_N,                          ! Record number of SCX
;  1620          2648  2         SCX_NS,                         ! Record number of successor of SCX
;  1621          2649  2         SCE:            REF BBLOCK;     ! Pointer to SCX entry
;  1622          2650  2
;  1623          2651  2
;  1624          2652  2 ! Search the characteristic index for the desired name.
;  1625          2653  2 !
;  1626          2654  2 SCX = READ_RECORD(SQH$K_RECNO);
;  1627          2655  2 SCX_N = .SCX[SQH$L_CHARACTERISTIC_LIST];
;  1628          2656  2 RELEASE_RECORD(SQH$K_RECNO);
;  1629          2657  2 WHILE .SCX_N NEQ 0 DO
;  1630          2658  3     BEGIN
;  1631          2659  3
;  1632          2660  3     ! Read the characteristic index record.
;  1633          2661  3     !
;  1634          2662  3     SCX = READ_RECORD(.SCX_N);
;  1635          2663  3
;  1636          2664  3
;  1637          2665  3     ! Search the characteristic index for the desired name.
;  1638          2666  3     !
;  1639          2667  3     SCE = SCX[SYH$T_DATA];
;  1640          2668  3     INCR SCE_N FROM 0 TO SCX$K_ENTRIES-1 DO
;  1641          2669  4         BEGIN
;  1642          2670  4         IF CH$RCHAR(SCE[SCX$T_NAME]) EQL 0
;  1643          2671  4         THEN
```

```
;  1644     2672  4            EXITLOOP
;  1645     2673  4        ELSE
;  1646     2674  5            BEGIN
;  1647     2675  5            CASE CH$COMPARE(
;  1648     2676  5                 CH$RCHAR(SCE[SCX$T_NAME]), SCE[SCX$T_NAME]+1,
;  1649     2677  5                 .DESC[SDSC_W_LENGTH], .DESC[SDSC_A_POINTER],
;  1650     2678  5                 %C' ')
;  1651     2679  5            FROM -1 TO 1 OF
;  1652     2680  5                SET
;  1653     2681  5
;  1654     2682  5                [-1]:
;  1655     2683  5                    SCE = .SCE + SCX$S_SCX;
;  1656     2684  5
;  1657     2685  5                [0]:
;  1658     2686  6                    BEGIN
;  1659     2687  6                    NUMBER = .SCE[SCX$B_NUMBER];
;  1660     2688  6                    RELEASE_RECORD(.SCX_N);
;  1661     2689  6                    RETURN TRUE;
;  1662     2690  6                    END;
;  1663     2691  5
;  1664     2692  5                [+1]:
;  1665     2693  6                    BEGIN
;  1666     2694  6                    RELEASE_RECORD(.SCX_N);
;  1667     2695  6                    RETURN FALSE;
;  1668     2696  6                    END;
;  1669     2697  5
;  1670     2698  5                TES;
;  1671     2699  4            END;
;  1672     2700  3        END;
;  1673     2701  3
;  1674     2702  3
;  1675     2703  3        ! Advance to the next index block.
;  1676     2704  3        !
;  1677     2705  3        SCX_NS = .SCX[SYM$L_LINK];
;  1678     2706  3        RELEASE_RECORD(.SCX_N);
;  1679     2707  3        SCX_N = .SCX_NS;
;  1680     2708  2        END;
;  1681     2709  2
;  1682     2710  2
;  1683     2711  2 FALSE
;  1684     2712  1 END;


                                07FC 00000          .ENTRY  FIND_CHARACTERISTIC, Save R2,R3,R4,R5,R6,-      ; 2615
                                                            R7,R8,R9,R10
                    5A 00000000G  EF   9E 00002     MOVAB   READ_RECORD, R10                                ;
                    54 00000000G  EF   9E 00009     MOVAB   RELEASE_RECORD, R4                              ;
                                  01   DD 00010     PUSHL   #1                                              ; 2654
                    6A            01   FB 00012     CALLS   #1, READ_RECORD                                 ;
                    58            50   D0 00015     MOVL    R0, SCX                                         ;
                    56      10    A8   D0 00018     MOVL    16(SCX), SCX_N                                  ; 2655
                                  01   DD 0001C     PUSHL   #1                                              ; 2656
                    64            01   FB 0001E     CALLS   #1, RELEASE_RECORD                              ;
                                  56   D5 00021 1$:  TSTL    SCX_N                                           ; 2657
```

```
                                        4F  13 00023          BEQL    7$                                              ;
                                        56  DD 00025          PUSHL   SCX_N                                           ; 2662
                                    6A  01  FB 00027          CALLS   #1, READ_RECORD
                                    58  50  D0 0002A          MOVL    R0, SCX
                                    55      0C  AB 9E 0002D   MOVAB   12(R8), SCE                                     ; 2667
                                        57  D4 00031          CLRL    SCE_N                                           ; 2668
                                        65  95 00033  2$:      TSTB    (SCE)                                           ; 2670
                                        30  13 00035          BEQL    6$
                                    51  65  9A 00037          MOVZBL  (SCE), R1                                       ; 2676
                                    50      04  AC D0 0003A    MOVL    DESC, R0                                        ; 2677
                60          20      01  A5  51  2D 0003E     CMPC5   R1, 1(SCE), #32, (R0), @2(R0)                   ; 2676
                                            02  B0  00044
                                        14  1A 00046          BGTRU   4$
                                        05  1E 00048          BGEQU   3$
                                    55      21  C0 0004A      ADDL2   #33, SCE                                        ; 2683
                                        14  11 0004D          BRB     5$
                                    5B      20  A5  9A 0004F  3$:      MOVZBL  32(SCE), NUMBER                         ; 2687
                                        56  DD 00053          PUSHL   SCX_N                                           ; 2688
                                    64  01  FB 00055          CALLS   #1, RELEASE_RECORD
                                    50  01  D0 00058          MOVL    #1, R0                                          ; 2689
                                        04  00058            RET
                                        56  DD 0005C  4$:      PUSHL   SCX_N                                           ; 2694
                                    64  01  FB 0005E          CALLS   #1, RELEASE_RECORD
                                        11  11 00061          BRB     7$                                              ; 2695
                        CC          57  0E  F3 00063  5$:      AOBLEQ  #14, SCE_N, 2$                                  ; 2668
                                    59  68  D0 00067  6$:      MOVL    (SCX), SCX_NS                                   ; 2705
                                        56  DD 0006A          PUSHL   SCX_N                                           ; 2706
                                    64  01  FB 0006C          CALLS   #1, RELEASE_RECORD
                                    56  59  D0 0006F          MOVL    SCX_NS, SCX_N                                   ; 2707
                                        AD  11 00072          BRB     1$                                              ; 2657
                                        50  D4 00074  7$:      CLRL    R0                                              ; 2712
                                            04  00076        RET

; Routine Size:  119 bytes,    Routine Base:  CODE + 09D9
```

```
;  1686          2713  1 GLOBAL ROUTINE FIND_FORM_NAME(DESC;SFM_N,SFM): L_OUTPUT_2=
;  1687          2714  1
;  1688          2715  1 !++
;  1689          2716  1 !
;  1690          2717  1 ! FUNCTIONAL DESCRIPTION:
;  1691          2718  1 !       This routine finds a form definition by name.
;  1692          2719  1 !
;  1693          2720  1 ! INPUT PARAMETERS:
;  1694          2721  1 !       DESC            - Short descriptor for form name.
;  1695          2722  1 !
;  1696          2723  1 ! IMPLICIT INPUTS:
;  1697          2724  1 !       NONE
;  1698          2725  1 !
;  1699          2726  1 ! OUTPUT PARAMETERS:
;  1700          2727  1 !       SFM_N           - Record number of SFM.
;  1701          2728  1 !       SFM             - Pointer to SFM.
;  1702          2729  1 !
;  1703          2730  1 ! IMPLICIT OUTPUTS:
;  1704          2731  1 !       NONE
;  1705          2732  1 !
;  1706          2733  1 ! ROUTINE VALUE:
;  1707          2734  1 !       True if the form is defined, false otherwise.
;  1708          2735  1 !
;  1709          2736  1 ! SIDE EFFECTS:
;  1710          2737  1 !       NONE
;  1711          2738  1 !
;  1712          2739  1 !--
;  1713          2740  1
;  1714          2741  2 BEGIN
;  1715          2742  2 MAP
;  1716          2743  2         DESC:           REF BBLOCK,     ! Short descriptor for form name
;  1717          2744  2         SFM:            REF BBLOCK;     ! Pointer to SFM
;  1718          2745  2 LOCAL
;  1719          2746  2         SFX:            REF BBLOCK,     ! Pointer to SFX
;  1720          2747  2         SFX_N,                          ! Record number of SFX
;  1721          2748  2         SFX_NS,                         ! Record number of successor of SFX
;  1722          2749  2         SFE:            REF BBLOCK;     ! Pointer to SFX entry
;  1723          2750  2
;  1724          2751  2
;  1725          2752  2 ! Search the form index for the desired name.
;  1726          2753  2 !
;  1727          2754  2 SFX = READ_RECORD(SQH$K_RECNO);
;  1728          2755  2 SFX_N = .SFX[SQH$L_FORM_INDEX_LIST];
;  1729          2756  2 RELEASE_RECORD(SQH$K_RECNO);
;  1730          2757  2 WHILE .SFX_N NEQ 0 DO
;  1731          2758  3     BEGIN
;  1732          2759  3
;  1733          2760  3     ! Read the form index record.
;  1734          2761  3     !
;  1735          2762  3     SFX = READ_RECORD(.SFX_N);
;  1736          2763  3
;  1737          2764  3
;  1738          2765  3     ! Search the form index for the desired name.
;  1739          2766  3     !
;  1740          2767  3     SFE = SFX[SYM$T_DATA];
;  1741          2768  3     INCR SFE_N FROM 0 TO SFX$K_ENTRIES-1 DO
;  1742          2769  4         BEGIN
```

```
;  1743     2770  4          IF CH$RCHAR(SFE[SFX$T_NAME]) EQL 0
;  1744     2771  4          THEN
;  1745     2772  4              EXITLOOP
;  1746     2773  4          ELSE
;  1747     2774  5              BEGIN
;  1748     2775  5              CASE CH$COMPARE(
;  1749     2776  5                  CH$RCHAR(SFE[SFX$T_NAME]), SFE[SFX$T_NAME]+1,
;  1750     2777  5                  .DESC[SDSC_W_LENGTH], .DESC[SDSC_A_POINTER],
;  1751     2778  5                  %C' ')
;  1752     2779  5              FROM -1 TO 1 OF
;  1753     2780  5                  SET
;  1754     2781  5
;  1755     2782  5                  [-1]:
;  1756     2783  5                      SFE = .SFE + SFX$S_SFX;
;  1757     2784  5
;  1758     2785  5                  [0]:
;  1759     2786  6                      BEGIN
;  1760     2787  6                      SFM = READ_RECORD(SFM_N = .SFE[SFX$L_FORM_LINK]);
;  1761     2788  6                      RELEASE_RECORD(.SFX_N);
;  1762     2789  6                      RETURN TRUE;
;  1763     2790  5                      END;
;  1764     2791  5
;  1765     2792  5                  [+1]:
;  1766     2793  6                      BEGIN
;  1767     2794  6                      RELEASE_RECORD(.SFX_N);
;  1768     2795  6                      RETURN FALSE;
;  1769     2796  5                      END;
;  1770     2797  5
;  1771     2798  5                  TES;
;  1772     2799  4              END;
;  1773     2800  3          END;
;  1774     2801  3
;  1775     2802  3
;  1776     2803  3          ! Advance to the next index block.
;  1777     2804  3          !
;  1778     2805  3          SFX_NS = .SFX[SYM$L_LINK];
;  1779     2806  3          RELEASE_RECORD(.SFX_N);
;  1780     2807  3          SFX_N = .SFX_NS;
;  1781     2808  2          END;
;  1782     2809  2
;  1783     2810  2
;  1784     2811  2  FALSE
;  1785     2812  1  END;


                                                03FC 00000          .ENTRY  FIND_FORM_NAME, Save R2,R3,R4,R5,R6,R7,R8,-      ; 2713
                                                                             R9
                                    54 00000000G  EF   9E 00002     MOVAB   RELEASE_RECORD, R4
                                                  01   DD 00009     PUSHL   #1                                               ; 2754
                                00000000G   EF    01   FB 0000B     CALLS   #1, READ_RECORD
                                            57    50   D0 00012     MOVL    R0, SFX
                                            56  34  A7   D0 00015     MOVL    52(SFX), SFX_N                               ; 2755
                                                  01   DD 00019     PUSHL   #1                                               ; 2756
                                            64    01   FB 0001B     CALLS   #1, RELEASE_RECORD
```

```
                                                 56  D5 0001E 1$:    TSTL    SFX_N                           ; 2757
                                                 60  13 00020        BEQL    7$
                                                 56  DD 00022        PUSHL   SFX_N                           ; 2762
                        00000000G                EF                  01  FB 00024        CALLS   #1, READ_RECORD
                                                 57                  50  D0 0002B        MOVL    R0, SFX
                                                 55          0C      A7  9E 0002E        MOVAB   12(R7), SFE     ; 2767
                                                                     5B  D4 00032        CLRL    SFE_N           ; 2768
                                                                     65  95 00034 2$:     TSTB    (SFE)           ; 2770
                                                                     3D  13 00036        BEQL    6$
                                                 51                  65  9A 00038        MOVZBL  (SFE), R1       ; 2776
                                                 50          04      AC  D0 0003B        MOVL    DESC, R0        ; 2777
                60              20      01      A5                  51  2D 0003F        CMPC5   R1, 1(SFE), #32, (R0), @2(R0)   ; 2776
                                                                 02  B0  00045
                                                                     21  1A 00047        BGTRU   4$
                                                                     05  1E 00049        BGEQU   3$
                                                 55                  28  C0 0004B        ADDL2   #40, SFE        ; 2783
                                                                     21  11 0004E        BRB     5$
                                                 5A          24      A5  D0 00050 3$:     MOVL    36(SFE), SFM_N  ; 2787
                                                                     5A  DD 00054        PUSHL   SFM_N
                        00000000G                EF                  01  FB 00056        CALLS   #1, READ_RECORD
                                                 59                  50  D0 0005D        MOVL    R0, SFM
                                                                     56  DD 00060        PUSHL   SFX_N           ; 2788
                                                 64                  01  FB 00062        CALLS   #1, RELEASE_RECORD
                                                 50                  01  D0 00065        MOVL    #1, R0          ; 2789
                                                                     1A  11 00068        BRB     8$
                                                                     56  DD 0006A 4$:     PUSHL   SFX_N           ; 2794
                                                 64                  01  FB 0006C        CALLS   #1, RELEASE_RECORD
                                                                     11  11 0006F        BRB     7$              ; 2795
                        BF                       5B                 0B  F3 00071 5$:     AOBLEQ  #11, SFE_N, 2$  ; 2768
                                                 58                  67  D0 00075 6$:     MOVL    (SFX), SFX_NS   ; 2805
                                                                     56  DD 00078        PUSHL   SFX_N           ; 2806
                                                 64                  01  FB 0007A        CALLS   #1, RELEASE_RECORD
                                                 56                  58  D0 0007D        MOVL    SFX_NS, SFX_N   ; 2807
                                                                     9C  11 00080        BRB     1$              ; 2757
                                                                     50  D4 00082 7$:     CLRL    R0              ; 2812
                                                 5B                  59  D0 00084 8$:     MOVL    R9, R11
                                                                         04 00087        RET

; Routine Size:  136 bytes,    Routine Base:  CODE + 0A50
```

```
;  1787     2813  1 GLOBAL ROUTINE FIND_FORM_NUMBER(NUMBER;SFM_N,SFM): L_OUTPUT_2=
;  1788     2814  1
;  1789     2815  1 !++
;  1790     2816  1 !
;  1791     2817  1 ! FUNCTIONAL DESCRIPTION:
;  1792     2818  1 !       This routine finds a form definition by number.
;  1793     2819  1 !
;  1794     2820  1 ! INPUT PARAMETERS:
;  1795     2821  1 !       NUMBER          - Form number.
;  1796     2822  1 !
;  1797     2823  1 ! IMPLICIT INPUTS:
;  1798     2824  1 !       NONE
;  1799     2825  1 !
;  1800     2826  1 ! OUTPUT PARAMETERS:
;  1801     2827  1 !       SFM_N           - Record number of SFM.
;  1802     2828  1 !       SFM             - Pointer to SFM.
;  1803     2829  1 !
;  1804     2830  1 ! IMPLICIT OUTPUTS:
;  1805     2831  1 !       NONE
;  1806     2832  1 !
;  1807     2833  1 ! ROUTINE VALUE:
;  1808     2834  1 !       True if the form is defined, false otherwise.
;  1809     2835  1 !
;  1810     2836  1 ! SIDE EFFECTS:
;  1811     2837  1 !       NONE
;  1812     2838  1 !
;  1813     2839  1 !--
;  1814     2840  1
;  1815     2841  2 BEGIN
;  1816     2842  2 MAP
;  1817     2843  2         SFM:            REF BBLOCK;     ! Pointer to SFM
;  1818     2844  2 LOCAL
;  1819     2845  2         SFX:            REF BBLOCK,     ! Pointer to SFX
;  1820     2846  2         SFX_N,                          ! Record number of SFX
;  1821     2847  2         SFX_NS,                         ! Record number of successor of SFX
;  1822     2848  2         SFE:            REF BBLOCK;     ! Pointer to SFX entry
;  1823     2849  2
;  1824     2850  2
;  1825     2851  2 ! Search the form index for the desired number.
;  1826     2852  2 !
;  1827     2853  2 SFX = READ_RECORD(SQH$K_RECNO);
;  1828     2854  2 SFX_N = .SFX[SQH$L_FORM_INDEX_LIST];
;  1829     2855  2 RELEASE_RECORD(SQH$K_RECNO);
;  1830     2856  2 WHILE .SFX_N NEQ 0 DO
;  1831     2857  3     BEGIN
;  1832     2858  3
;  1833     2859  3     ! Read the form index record.
;  1834     2860  3     !
;  1835     2861  3     SFX = READ_RECORD(.SFX_N);
;  1836     2862  3
;  1837     2863  3
;  1838     2864  3     ! Search the form index for the desired number.
;  1839     2865  3     !
;  1840     2866  3     SFE = SFX[SYM$T_DATA];
;  1841     2867  3     INCR SFE_N FROM 0 TO $FX$K_ENTRIES-1 DO
;  1842     2868  4         BEGIN
;  1843     2869  4         IF CH$RCHAR(SFE[SFX$T_NAME]) EQL 0
```

```
;  1844     2870  4          THEN
;  1845     2871  4              EXITLOOP
;  1846     2872  4          ELSE
;  1847     2873  4              IF .NUMBER EQL .SFE[SFX$L_NUMBER]
;  1848     2874  4              THEN
;  1849     2875  5                  BEGIN
;  1850     2876  5                  SFM = READ_RECORD(SFM_N = .SFE[SFX$L_FORM_LINK]);
;  1851     2877  5                  RELEASE_RECORD(.SFX_N);
;  1852     2878  5                  RETURN TRUE;
;  1853     2879  5                  END
;  1854     2880  4              ELSE
;  1855     2881  4                  SFE = .SFE + SFX$S_SFX;
;  1856     2882  3          END;
;  1857     2883  3
;  1858     2884  3
;  1859     2885  3      ! Advance to the next index block.
;  1860     2886  3      !
;  1861     2887  3      SFX_NS = .SFX[SYM$L_LINK];
;  1862     2888  3      RELEASE_RECORD(.SFX_N);
;  1863     2889  3      SFX_N = .SFX_NS;
;  1864     2890  2      END;
;  1865     2891  2
;  1866     2892  2
;  1867     2893  2  FALSE
;  1868     2894  1  END;


                                 01FC 00000          .ENTRY  FIND_FORM_NUMBER, Save R2,R3,R4,R5,R6,R7,R8 ; 2813
                     58 00000000G   EF  9E 00002     MOVAB   READ_RECORD, R8
                     57 00000000G   EF  9E 00009     MOVAB   RELEASE_RECORD, R7
                                    01  DD 00010     PUSHL   #1                                          ; 2853
                     68             01  FB 00012     CALLS   #1, READ_RECORD
                     54             50  D0 00015     MOVL    R0, SFX
                     5B       34    A4  D0 00018     MOVL    52(SFX), SFX_N                              ; 2854
                                    01  DD 0001C     PUSHL   #1                                          ; 2855
                     67             01  FB 0001E     CALLS   #1, RELEASE_RECORD
                                    5B  D5 00021 1$:  TSTL    SFX_N                                       ; 2856
                                    43  13 00023     BEQL    5$
                                    5B  DD 00025     PUSHL   SFX_N                                       ; 2861
                     68             01  FB 00027     CALLS   #1, READ_RECORD
                     54             50  D0 0002A     MOVL    R0, SFX
                     52       0C    A4  9E 0002D     MOVAB   12(R4), SFE                                 ; 2866
                                    53  D4 00031     CLRL    SFE_N                                       ; 2867
                                    62  95 00033 2$:  TSTB    (SFE)                                       ; 2869
                                    24  13 00035     BEQL    4$
                20   A2       04    AC  D1 00037     CMPL    NUMBER, 32(SFE)                             ; 2873
                                    16  12 0003C     BNEQ    3$
                     5A       24    A2  D0 0003E     MOVL    36(SFE), SFM_N                              ; 2876
                                    5A  DD 00042     PUSHL   SFM_N
                     68             01  FB 00044     CALLS   #1, READ_RECORD
                     56             50  D0 00047     MOVL    R0, SFM
                                    5B  DD 0004A     PUSHL   SFX_N                                       ; 2877
                     67             01  FB 0004C     CALLS   #1, RELEASE_RECORD
                     50             01  D0 0004F     MOVL    #1, R0                                      ; 2878
```

```
                        16  11 00052            BRB     6$                                  ;
                52      28  C0 00054 3$:        ADDL2   #40, SFE                            ; 2881
              D8 53     0B  F3 00057            AOBLEQ  #11, SFE_N, 2$                      ; 2867
                55      64  D0 0005B 4$:        MOVL    (SFX), SFX_NS                       ; 2887
                        5B  DD 0005E            PUSHL   SFX_N                               ; 2888
                67      01  FB 00060            CALLS   #1, RELEASE_RECORD                  ;
                5B      55  D0 00063            MOVL    SFX_NS, SFX_N                       ; 2889
                        B9  11 00066            BRB     1$                                  ; 2856
                        50  D4 00068 5$:        CLRL    R0                                  ; 2894
                5B      56  D0 0006A 6$:        MOVL    R6, R11                             ;
                        04  0006D               RET                                         ;

; Routine Size:  110 bytes,    Routine Base:  CODE + 0AD8
```

```
;1870   2895  1 ROUTINE FIND_FORM_REFERENCES_J(SFM_NF,SJH_NO)=
;1871   2896  1
;1872   2897  1 !++
;1873   2898  1 !
;1874   2899  1 ! FUNCTIONAL DESCRIPTION:
;1875   2900  1 !       This routine finds references to a specified form in a list of jobs.
;1876   2901  1 !
;1877   2902  1 ! INPUT PARAMETERS:
;1878   2903  1 !       SFM_NF          - Record number of SFM.
;1879   2904  1 !       SJH_NO          - Record number of SJH.
;1880   2905  1 !
;1881   2906  1 ! IMPLICIT INPUTS:
;1882   2907  1 !       NONE
;1883   2908  1 !
;1884   2909  1 ! OUTPUT PARAMETERS:
;1885   2910  1 !       NONE
;1886   2911  1 !
;1887   2912  1 ! IMPLICIT OUTPUTS:
;1888   2913  1 !       NONE
;1889   2914  1 !
;1890   2915  1 ! ROUTINE VALUE:
;1891   2916  1 !       True if any references were found, false otherwise.
;1892   2917  1 !
;1893   2918  1 ! SIDE EFFECTS:
;1894   2919  1 !       NONE
;1895   2920  1 !
;1896   2921  1 !--
;1897   2922  1
;1898   2923  2 BEGIN
;1899   2924  2 LOCAL
;1900   2925  2         SJH_NS,                         ! Record number of successor of SJH
;1901   2926  2         SJH_N,                          ! Record number of SJH
;1902   2927  2         SJH:            REF BBLOCK;     ! Pointer to SJH
;1903   2928  2
;1904   2929  2
;1905   2930  2 SJH_N = .SJH_NO;
;1906   2931  2 WHILE .SJH_N NEQ 0 DO
;1907   2932  3     BEGIN
;1908   2933  3     SJH = READ_RECORD(.SJH_N);
;1909   2934  3
;1910   2935  3
;1911   2936  3     IF .SJH[SJH$L_FORM_LINK] EQL .SFM_NF
;1912   2937  3     THEN
;1913   2938  4         BEGIN
;1914   2939  4         RELEASE_RECORD(.SJH_N);
;1915   2940  4         RETURN TRUE;
;1916   2941  3         END;
;1917   2942  3
;1918   2943  3
;1919   2944  3     SJH_NS = .SJH[SYM$L_LINK];
;1920   2945  3     RELEASE_RECORD(.SJH_N);
;1921   2946  3     SJH_N = .SJH_NS;
;1922   2947  2     END;
;1923   2948  2
;1924   2949  2
;1925   2950  2 FALSE
;1926   2951  1 END;
```

```
                                        003C 00000 FIND_FORM_REFERENCES_J:
                                                   .WORD   Save R2,R3,R4,R5                ; 2895
                          55 00000000G  EF  9E 00002       MOVAB   RELEASE_RECORD, R5
                          52       08   AC  D0 00009       MOVL    SJH_NO, SJH_N           ; 2930
                                        2A  13 0000D 1$:    BEQL    3$                      ; 2931
                                        52  DD 0000F       PUSHL   SJH_N                   ; 2933
                 00000000G  EF          01  FB 00011       CALLS   #1, READ_RECORD
                          53            50  D0 00018       MOVL    R0, SJH
                       04 AC      00FC  C3  D1 0001B       CMPL    252(SJH), SFM_NF        ; 2936
                                        09  12 00021       BNEQ    2$
                                        52  DD 00023       PUSHL   SJH_N                   ; 2939
                          65            01  FB 00025       CALLS   #1, RELEASE_RECORD
                          50            01  D0 00028       MOVL    #1, R0                  ; 2940
                                            04 0002B       RET
                          54            63  D0 0002C 2$:    MOVL    (SJH), SJH_NS           ; 2944
                                        52  DD 0002F       PUSHL   SJH_N                   ; 2945
                          65            01  FB 00031       CALLS   #1, RELEASE_RECORD
                          52            54  D0 00034       MOVL    SJH_NS, SJH_N           ; 2946
                                        D4  11 00037       BRB     1$                      ; 2931
                                        50  D4 00039 3$:    CLRL    R0                      ; 2951
                                            04 0003B       RET

; Routine Size:  60 bytes,    Routine Base:  CODE + 0B46
```

```
;  1928     2952  1 GLOBAL ROUTINE FIND_FORM_REFERENCES(SFM_NF)=
;  1929     2953  1
;  1930     2954  1 !++
;  1931     2955  1 !
;  1932     2956  1 ! FUNCTIONAL DESCRIPTION:
;  1933     2957  1 !       This routine finds references to a specified form anywhere in the queue.
;  1934     2958  1 !
;  1935     2959  1 ! INPUT PARAMETERS:
;  1936     2960  1 !       SFM_NF          - Record number of SFM.
;  1937     2961  1 !
;  1938     2962  1 ! IMPLICIT INPUTS:
;  1939     2963  1 !       NONE
;  1940     2964  1 !
;  1941     2965  1 ! OUTPUT PARAMETERS:
;  1942     2966  1 !       NONE
;  1943     2967  1 !
;  1944     2968  1 ! IMPLICIT OUTPUTS:
;  1945     2969  1 !       NONE
;  1946     2970  1 !
;  1947     2971  1 ! ROUTINE VALUE:
;  1948     2972  1 !       NONE
;  1949     2973  1 !
;  1950     2974  1 ! SIDE EFFECTS:
;  1951     2975  1 !       NONE
;  1952     2976  1 !
;  1953     2977  1 !--
;  1954     2978  1
;  1955     2979  2 BEGIN
;  1956     2980  2 LOCAL
;  1957     2981  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  1958     2982  2         SQX:            REF BBLOCK,     ! Pointer to SQX
;  1959     2983  2         SQX_N,                          ! Record number of SQX
;  1960     2984  2         SQX_NS,                         ! Record number of successor of SQX
;  1961     2985  2         SQE:            REF BBLOCK,     ! Pointer to SQX entry
;  1962     2986  2         SMQ_N,                          ! Record number of SMQ
;  1963     2987  2         SMQ:            REF BBLOCK;     ! Pointer to SMQ
;  1964     2988  2
;  1965     2989  2
;  1966     2990  2 ! Read the queue header.
;  1967     2991  2 !
;  1968     2992  2 SQH = READ_RECORD(SQH$K_RECNO);
;  1969     2993  2
;  1970     2994  2
;  1971     2995  2 ! Search for form references in each job list linked from the queue header.
;  1972     2996  2 !
;  1973     2997  2 IF FIND_FORM_REFERENCES_J(.SFM_NF, .SQH[SQH$L_OPEN_LIST])
;  1974     2998  2 OR FIND_FORM_REFERENCES_J(.SFM_NF, .SQH[SQH$L_PENDING_BATCH_LIST])
;  1975     2999  2 OR FIND_FORM_REFERENCES_J(.SFM_NF, .SQH[SQH$L_PENDING_PRINT_LIST])
;  1976     3000  2 OR FIND_FORM_REFERENCES_J(.SFM_NF, .SQH[SQH$L_TIMER_LIST])
;  1977     3001  2 THEN
;  1978     3002  3     BEGIN
;  1979     3003  3     RELEASE_RECORD(SQH$K_RECNO);
;  1980     3004  3     RETURN TRUE;
;  1981     3005  3     END;
;  1982     3006  2
;  1983     3007  2
;  1984     3008  2 ! Loop over all queue headers.
```

```
;  1985          3009  2 !
;  1986          3010  2 SQX_N = .SQH[SQH$L_QUEUE_INDEX_LIST];
;  1987          3011  2 RELEASE_RECORD(SQH$K_RECNO);
;  1988          3012  2 WHILE .SQX_N NEQ 0 DO
;  1989          3013  2     BEGIN
;  1990          3014  2
;  1991          3015  2     ! Read the queue index record.
;  1992          3016  2     !
;  1993          3017  2     SQX = READ_RECORD(.SQX_N);
;  1994          3018  2
;  1995          3019  2
;  1996          3020  2     ! Search the queue index.
;  1997          3021  2     !
;  1998          3022  2     SQE = SQX[SYM$T_DATA];
;  1999          3023  3     INCR SQE_N FROM 0 TO SQX$K_ENTRIES-1 DO
;  2000          3024  4         BEGIN
;  2001          3025  4         IF CH$RCHAR(SQE[SQX$T_NAME]) EQL 0
;  2002          3026  4         THEN
;  2003          3027  4             EXITLOOP
;  2004          3028  4         ELSE
;  2005          3029  5             BEGIN
;  2006          3030  5
;  2007          3031  5             ! Read the queue header.
;  2008          3032  5             !
;  2009          3033  5             SMQ = READ_RECORD(SMQ_N = .SQE[SQX$L_QUEUE_LINK]);
;  2010          3034  5
;  2011          3035  5
;  2012          3036  5             ! Search for form references in the queue header and in each job
;  2013          3037  5             ! list linked from the queue header.
;  2014          3038  5             !
;  2015          3039  5             IF .SMQ[SMQ$L_FORM_LINK] EQL .SFM_NF
;  2016          3040  5             OR FIND_FORM_REFERENCES_J(.SFM_NF, .SMQ[SMQ$L_CURRENT_LIST])
;  2017          3041  5             OR FIND_FORM_REFERENCES_J(.SFM_NF, .SMQ[SMQ$L_HOLD_LIST])
;  2018          3042  5             THEN
;  2019          3043  6                 BEGIN
;  2020          3044  6                 RELEASE_RECORD(.SMQ_N);
;  2021          3045  6                 RELEASE_RECORD(.SQX_N);
;  2022          3046  6                 RETURN TRUE;
;  2023          3047  6                 END;
;  2024          3048  5
;  2025          3049  5
;  2026          3050  5             ! Release the queue header.
;  2027          3051  5             !
;  2028          3052  5             RELEASE_RECORD(.SMQ_N);
;  2029          3053  4             END;
;  2030          3054  4
;  2031          3055  4
;  2032          3056  4         SQE = .SQE + SQX$S_SQX;
;  2033          3057  4         END;
;  2034          3058  3
;  2035          3059  3
;  2036          3060  3     ! Advance to the next index block.
;  2037          3061  3     !
;  2038          3062  3     SQX_NS = .SQX[SYM$L_LINK];
;  2039          3063  3     RELEASE_RECORD(.SQX_N);
;  2040          3064  3     SQX_N = .SQX_NS;
;  2041          3065  2     END;
```

```
;  2042          3066  2
;  2043          3067  2
;  2044          3068  2 FALSE
;  2045          3069  1 END;


                                        0FFC 00000          .ENTRY  FIND_FORM_REFERENCES, Save R2,R3,R4,R5,R6,- ; 2952
                                                                    R7,R8,R9,R10,R11
                                5B      BF   AF   9E 00002  MOVAB   FIND_FORM_REFERENCES_J, R11
                                5A 00000000G EF   9E 00006  MOVAB   RELEASE_RECORD, R10
                                             01   DD 00009  PUSHL   #1                                          ; 2992
                      00000000G EF                01   FB 0000C  CALLS   #1, READ_RECORD
                                52                50   D0 00013  MOVL    R0, SQH
                                          4C    A2   DD 00016  PUSHL   76(SQH)                                  ; 2997
                                54        04    AC   D0 00019  MOVL    SFM_NF, R4
                                                   54   DD 0001D  PUSHL   R4
                                6B                 02   FB 0001F  CALLS   #2, FIND_FORM_REFERENCES_J
                                21                 50   E8 00022  BLBS    R0, 1$
                                          54    A2   DD 00025  PUSHL   84(SQH)                                  ; 2998
                                                   54   DD 00028  PUSHL   R4
                                6B                 02   FB 0002A  CALLS   #2, FIND_FORM_REFERENCES_J
                                16                 50   E8 0002D  BLBS    R0, 1$
                                          5C    A2   DD 00030  PUSHL   92(SQH)                                  ; 2999
                                                   54   DD 00033  PUSHL   R4
                                6B                 02   FB 00035  CALLS   #2, FIND_FORM_REFERENCES_J
                                0B                 50   E8 00038  BLBS    R0, 1$
                                          68    A2   DD 0003B  PUSHL   104(SQH)                                 ; 3000
                                                   54   DD 0003E  PUSHL   R4
                                6B                 02   FB 00040  CALLS   #2, FIND_FORM_REFERENCES_J
                                04                 50   E9 00043  BLBC    R0, 2$
                                                   01   DD 00046  PUSHL   #1                                    ; 3003
                                                   56   11 00048  BRB     6$
                                57        64    A2   D0 0004A  2$:  MOVL    100(SQH), SQX_N                    ; 3010
                                                   01   DD 0004E  PUSHL   #1                                    ; 3011
                                6A                 01   FB 00050  CALLS   #1, RELEASE_RECORD
                                                   57   D5 00053  3$:  TSTL    SQX_N                           ; 3012
                                                   69   13 00055  BEQL    9$
                                                   57   DD 00057  PUSHL   SQX_N                                 ; 3017
                      00000000G EF                01   FB 00059  CALLS   #1, READ_RECORD
                                59                50   D0 00060  MOVL    R0, SQX
                                53        0C    A9   9E 00063  MOVAB   12(R9), SQE                            ; 3022
                                                   56   D4 00067  CLRL    SQE_N                               ; 3023
                                                   63   95 00069  4$:  TSTB    (SQE)                           ; 3025
                                                   46   13 0006B  BEQL    8$
                                55        24    A3   D0 0006D  MOVL    36(SQE), SMQ_N                      ; 3033
                                                   55   DD 00071  PUSHL   SMQ_N
                      00000000G EF                01   FB 00073  CALLS   #1, READ_RECORD
                                52                50   D0 0007A  MOVL    R0, SMQ
                                54        70    A2   D1 0007D  CMPL    112(SMQ), R4                        ; 3039
                                                   16   13 00081  BEQL    5$
                                          48    A2   DD 00083  PUSHL   72(SMQ)                                  ; 3040
                                                   54   DD 00086  PUSHL   R4
                                6B                 02   FB 00088  CALLS   #2, FIND_FORM_REFERENCES_J
                                0B                 50   E8 0008B  BLBS    R0, 5$
```

```
                     78  A2  DD 00091          PUSHL   120(SMQ)                          ; 3041
                         54  DD 00094          PUSHL   R4                              ;
                     6B      02  FB 00096      CALLS   #2, FIND_FORM_REFERENCES_J      ;
                     0E      50  E9 00099      BLBC    R0, 7$                          ;
                         55  DD 0009C 5$:      PUSHL   SMQ_N                           ; 3044
                     6A      01  FB 0009E      CALLS   #1, RELEASE_RECORD              ;
                         57  DD 000A1          PUSHL   SQX_N                           ; 3045
                     6A      01  FB 000A3 6$:  CALLS   #1, RELEASE_RECORD              ;
                     50      01  D0 000A6      MOVL    #1, R0                          ; 3046
                             04  000A9        RET                                     ;
                         55  DD 000AA 7$:      PUSHL   SMQ_N                           ; 3052
                     6A      01  FB 000AC      CALLS   #1, RELEASE_RECORD              ;
                     53      28  C0 000AF      ADDL2   #40, SQE                        ; 3056
                B6   56      08  F3 000B2      AOBLEQ  #11, SQE_N, 4$                  ; 3023
                     58      69  D0 000B6 8$:  MOVL    (SQX), SQX_NS                   ; 3062
                         57  DD 000B9          PUSHL   SQX_N                           ; 3063
                     6A      01  FB 000BB      CALLS   #1, RELEASE_RECORD              ;
                     57      58  D0 000BE      MOVL    SQX_NS, SQX_N                   ; 3064
                         93  11  000C1        BRB     3$                              ; 3012
                         50  D4  000C3 9$:     CLRL    R0                              ; 3069
                             04  000C5        RET                                     ;

; Routine Size:  198 bytes,    Routine Base:  CODE + 0B82
```

```
;  2047     3070  1 ROUTINE FIND_QUEUE_REFERENCES_J(SMQ_NF,SJH_NO)=
;  2048     3071  1
;  2049     3072  1 !++
;  2050     3073  1 !
;  2051     3074  1 ! FUNCTIONAL DESCRIPTION:
;  2052     3075  1 !       This routine finds references to a specified queue in a list of jobs.
;  2053     3076  1 !
;  2054     3077  1 ! INPUT PARAMETERS:
;  2055     3078  1 !       SMQ_NF          - Record number of SMQ.
;  2056     3079  1 !       SJH_NO          - Record number of SJH.
;  2057     3080  1 !
;  2058     3081  1 ! IMPLICIT INPUTS:
;  2059     3082  1 !       NONE
;  2060     3083  1 !
;  2061     3084  1 ! OUTPUT PARAMETERS:
;  2062     3085  1 !       NONE
;  2063     3086  1 !
;  2064     3087  1 ! IMPLICIT OUTPUTS:
;  2065     3088  1 !       NONE
;  2066     3089  1 !
;  2067     3090  1 ! ROUTINE VALUE:
;  2068     3091  1 !       True if any references were found, false otherwise.
;  2069     3092  1 !
;  2070     3093  1 ! SIDE EFFECTS:
;  2071     3094  1 !       NONE
;  2072     3095  1 !
;  2073     3096  1 !--
;  2074     3097  1
;  2075     3098  2 BEGIN
;  2076     3099  2 LOCAL
;  2077     3100  2         SJH_NS,                         ! Record number of successor of SJH
;  2078     3101  2         SJH_N,                          ! Record number of SJH
;  2079     3102  2         SJH:            REF BBLOCK;     ! Pointer to SJH
;  2080     3103  2
;  2081     3104  2
;  2082     3105  2 SJH_N = .SJH_NO;
;  2083     3106  2 WHILE .SJH_N NEQ 0 DO
;  2084     3107  2     BEGIN
;  2085     3108  2     SJH = READ_RECORD(.SJH_N);
;  2086     3109  2
;  2087     3110  2
;  2088     3111  2     IF .SJH[SJH$L_LOG_QUEUE_LINK] EQL .SMQ_NF
;  2089     3112  2     OR .SJH[SJH$L_QUEUE_LINK] EQL .SMQ_NF
;  2090     3113  2     OR .SJH[SJH$L_REQUEUE_QUEUE_LINK] EQL .SMQ_NF
;  2091     3114  2     THEN
;  2092     3115  2         BEGIN
;  2093     3116  2         RELEASE_RECORD(.SJH_N);
;  2094     3117  2         RETURN TRUE;
;  2095     3118  2         END;
;  2096     3119  2
;  2097     3120  2
;  2098     3121  2     SJH_NS = .SJH[SYM$L_LINK];
;  2099     3122  2     RELEASE_RECORD(.SJH_N);
;  2100     3123  2     SJH_N = .SJH_NS;
;  2101     3124  2     END;
;  2102     3125  2
;  2103     3126  2
```

```
;  2104          3127  2 FALSE
;  2105          3128  1 END;


                                                003C 00000 FIND_QUEUE_REFERENCES J:
                                                           .WORD   Save R2,R3,R4,R5                ; 3070
                                    55 00000000G   EF  9E 00002    MOVAB   RELEASE_RECORD, R5      ;
                                    53       08    AC  D0 00009    MOVL    SJH_N0, SJH_N           ; 3105
                                                   3A  13 0000D 1$:  BEQL   4$                     ; 3106
                                                   53  DD 0000F    PUSHL   SJH_N                   ; 3108
                                00000000G   EF     01  FB 00011    CALLS   #1, READ_RECORD         ;
                                            52     50  D0 00018    MOVL    R0, SJH                 ;
                                    04   AC  0104  C2  D1 0001B    CMPL    260(SJH), SMQ_NF        ; 3111
                                                   10  13 00021    BEQL    2$                      ;
                                    04   AC  0134  C2  D1 00023    CMPL    308(SJH), SMQ_NF        ; 3112
                                                   08  13 00029    BEQL    2$                      ;
                                    04   AC  0138  C2  D1 0002B    CMPL    312(SJH), SMQ_NF        ; 3113
                                                   09  12 00031    BNEQ    3$                      ;
                                                   53  DD 00033 2$:  PUSHL   SJH_N                   ; 3116
                                            65     01  FB 00035    CALLS   #1, RELEASE_RECORD      ;
                                            50     01  D0 00038    MOVL    #1, R0                  ; 3117
                                                   04  0003B      RET                             ;
                                            54     62  D0 0003C 3$:  MOVL    (SJH), SJH_NS           ; 3121
                                                   53  DD 0003F    PUSHL   SJH_N                   ; 3122
                                            65     01  FB 00041    CALLS   #1, RELEASE_RECORD      ;
                                            53     54  D0 00044    MOVL    SJH_NS, SJH_N           ; 3123
                                                   C4  11 00047    BRB     1$                      ; 3106
                                                   50  D4 00049 4$:  CLRL    R0                      ; 3128
                                                   04  0004B      RET                             ;

; Routine Size:  76 bytes,    Routine Base:  CODE + 0C48
```

```
;  2107     3129  1 GLOBAL ROUTINE FIND_QUEUE_REFERENCES(SMQ_NF)=
;  2108     3130  1
;  2109     3131  1 !++
;  2110     3132  1 !
;  2111     3133  1 ! FUNCTIONAL DESCRIPTION:
;  2112     3134  1 !       This routine finds references to a specified queue anywhere in the
;  2113     3135  1 !       queue.
;  2114     3136  1 !
;  2115     3137  1 ! INPUT PARAMETERS:
;  2116     3138  1 !       SMQ_NF          - Record number of SMQ.
;  2117     3139  1 !
;  2118     3140  1 ! IMPLICIT INPUTS:
;  2119     3141  1 !       NONE
;  2120     3142  1 !
;  2121     3143  1 ! OUTPUT PARAMETERS:
;  2122     3144  1 !       NONE
;  2123     3145  1 !
;  2124     3146  1 ! IMPLICIT OUTPUTS:
;  2125     3147  1 !       NONE
;  2126     3148  1 !
;  2127     3149  1 ! ROUTINE VALUE:
;  2128     3150  1 !       NONE
;  2129     3151  1 !
;  2130     3152  1 ! SIDE EFFECTS:
;  2131     3153  1 !       NONE
;  2132     3154  1 !
;  2133     3155  1 !--
;  2134     3156  1
;  2135     3157  2 BEGIN
;  2136     3158  2 LOCAL
;  2137     3159  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  2138     3160  2         SQX:            REF BBLOCK,     ! Pointer to SQX
;  2139     3161  2         SQX_N,                          ! Record number of SQX
;  2140     3162  2         SQX_NS,                         ! Record number of successor of SQX
;  2141     3163  2         SQE:            REF BBLOCK,     ! Pointer to SQX entry
;  2142     3164  2         SMQ_N,                          ! Record number of SMQ
;  2143     3165  2         SMQ:            REF BBLOCK;     ! Pointer to SMQ
;  2144     3166  2
;  2145     3167  2
;  2146     3168  2 ! Read the queue header.
;  2147     3169  2 !
;  2148     3170  2 SQH = READ_RECORD(SQH$K_RECNO);
;  2149     3171  2
;  2150     3172  2
;  2151     3173  2 ! Search for queue references in each job list linked from the queue header.
;  2152     3174  2 !
;  2153     3175  2 IF FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SQH[SQH$L_OPEN_LIST])
;  2154     3176  2 OR FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SQH[SQH$L_PENDING_BATCH_LIST])
;  2155     3177  2 OR FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SQH[SQH$L_PENDING_PRINT_LIST])
;  2156     3178  2 OR FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SQH[SQH$L_TIMER_LIST])
;  2157     3179  2 THEN
;  2158     3180  3     BEGIN
;  2159     3181  3     RELEASE_RECORD(SQH$K_RECNO);
;  2160     3182  3     RETURN TRUE;
;  2161     3183  3     END;
;  2162     3184  2
;  2163     3185  2
```

```
;  2164          3186  2 ! Loop over all queue headers.
;  2165          3187  2 !
;  2166          3188  2 SQX_N = .SQH[SQH$L_QUEUE_INDEX_LIST];
;  2167          3189  2 RELEASE_RECORD(SQH$K_RECNO);
;  2168          3190  2 WHILE .SQX_N NEQ 0 DO
;  2169          3191  3     BEGIN
;  2170          3192  3
;  2171          3193  3     ! Read the queue index record.
;  2172          3194  3     !
;  2173          3195  3     SQX = READ_RECORD(.SQX_N);
;  2174          3196  3
;  2175          3197  3
;  2176          3198  3     ! Search the queue index.
;  2177          3199  3     !
;  2178          3200  3     SQE = SQX[SYM$T_DATA];
;  2179          3201  3     INCR SQE_N FROM 0 TO SQX$K_ENTRIES-1 DO
;  2180          3202  4         BEGIN
;  2181          3203  4         IF CH$RCHAR(SQE[SQX$T_NAME]) EQL 0
;  2182          3204  4         THEN
;  2183          3205  4             EXITLOOP
;  2184          3206  4         ELSE
;  2185          3207  5             BEGIN
;  2186          3208  5
;  2187          3209  5             ! Read the queue header.
;  2188          3210  5             !
;  2189          3211  5             SMQ = READ_RECORD(SMQ_N = .SQE[SQX$L_QUEUE_LINK]);
;  2190          3212  5
;  2191          3213  5
;  2192          3214  5             ! Search for queue references in the queue header and in each job
;  2193          3215  5             ! list linked from the queue header.
;  2194          3216  5             !
;  2195          3217  5             IF .SMQ[SMQ$L_ASSIGNED_QUEUE_LINK] EQL .SMQ_NF
;  2196          3218  5             OR FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SMQ[SMQ$L_CURRENT_LIST])
;  2197          3219  5             OR FIND_QUEUE_REFERENCES_J(.SMQ_NF, .SMQ[SMQ$L_HOLD_LIST])
;  2198          3220  5             THEN
;  2199          3221  6                 BEGIN
;  2200          3222  6                 RELEASE_RECORD(.SMQ_N);
;  2201          3223  6                 RELEASE_RECORD(.SQX_N);
;  2202          3224  6                 RETURN TRUE;
;  2203          3225  5                 END;
;  2204          3226  5
;  2205          3227  5
;  2206          3228  5             ! Search for generic target references to the queue.
;  2207          3229  5             !
;  2208          3230  5             IF .SMQ[SMQ$L_GENERIC_TARGET] NEQ 0
;  2209          3231  5             THEN
;  2210          3232  6                 BEGIN
;  2211          3233  6                 LOCAL
;  2212          3234  6                     AUX_N,                      ! Number of auxiliary record
;  2213          3235  6                     AUX:        REF BBLOCK;     ! Pointer to auxiliary record
;  2214          3236  6
;  2215          3237  6                 AUX = READ_RECORD(AUX_N = .SMQ[SMQ$L_GENERIC_TARGET]);
;  2216          3238  6                 DECR N FROM .VECTOR[AUX[SYM$T_DATA], 0] TO 1 DO
;  2217          3239  7                     BEGIN
;  2218          3240  7                     IF .VECTOR[AUX[SYM$T_DATA], .N] EQL .SMQ_NF
;  2219          3241  7                     THEN
;  2220          3242  8                         BEGIN
```

```
;  2221          3243  8                        RELEASE_RECORD(.AUX_N);
;  2222          3244  8                        RELEASE_RECORD(.SMQ_N);
;  2223          3245  8                        RELEASE_RECORD(.SQX_N);
;  2224          3246  8                        RETURN TRUE;
;  2225          3247  7                        END;
;  2226          3248  6                    END;
;  2227          3249  6                RELEASE_RECORD(.AUX_N);
;  2228          3250  5                END;
;  2229          3251  5
;  2230          3252  5
;  2231          3253  5                ! Release the queue header.
;  2232          3254  5                !
;  2235          3255  5                RELEASE_RECORD(.SMQ_N);
;  2234          3256  4                END;
;  2235          3257  4
;  2236          3258  4
;  2237          3259  4            SQE = .SQE + SQX$S_SQX;
;  2238          3260  4            END;
;  2239          3261  3
;  2240          3262  3
;  2241          3263  3        ! Advance to the next index block.
;  2242          3264  3        !
;  2243          3265  3        SQX_NS = .SQX[SYM$L_LINK];
;  2244          3266  3        RELEASE_RECORD(.SQX_N);
;  2245          3267  3        SQX_N = .SQX_NS;
;  2246          3268  3        END;
;  2247          3269  2
;  2248          3270  2
;  2249          3271  2    FALSE
;  2250          3272  1    END;


                                    0FFC 00000      .ENTRY  FIND_QUEUE_REFERENCES, Save R2,R3,R4,R5,R6,-  ; 3129
                                                            R7,R8,R9,RT0,R11
                          5E                  04 C2 00002   SUBL2   #4, SP
                                              01 DD 00005   PUSHL   #1                                    ; 3170
                00000000G EF                  01 FB 00007   CALLS   #1, READ_RECORD
                          52                  50 D0 0000E   MOVL    R0, SQH
                                        4C    A2 DD 00011   PUSHL   76(SQH)                               ; 3175
                          56            04    AC D0 00014   MOVL    SMQ_NF, R6
                                              56 DD 00018   PUSHL   R6
                       96 AF                  02 FB 0001A   CALLS   #2, FIND_QUEUE_REFERENCES_J
                          26                  50 E8 0001E   BLBS    R0, 1$
                                        54    A2 DD 00021   PUSHL   84(SQH)                               ; 3176
                                              56 DD 00024   PUSHL   R6
                       8A AF                  02 FB 00026   CALLS   #2, FIND_QUEUE_REFERENCES_J
                          1A                  50 E8 0002A   BLBS    R0, 1$
                                        5C    A2 DD 0002D   PUSHL   92(SQH)                               ; 3177
                                              56 DD 00030   PUSHL   R6
                     FF7D CF                  02 FB 00032   CALLS   #2, FIND_QUEUE_REFERENCES_J
                          0D                  50 E8 00037   BLBS    R0, 1$
                                        68    A2 DD 0003A   PUSHL   104(SQH)                              ; 3178
                                              56 DD 0003D   PUSHL   R6
                     FF70 CF                  02 FB 0003F   CALLS   #2, FIND_QUEUE_REFERENCES_J
```

```
                        05              50  E9  00044             BLBC    R0, 2$                                  ;
                                        01  DD  00047 1$:          PUSHL   #1                                      ; 3181
                                    0094  31  00049               BRW     9$                                      ;
                        58          64  A2  DO  0004C 2$:          MOVL    100(SQH), SQX_N                         ; 3188
                                        01  DD  00050             PUSHL   #1                                      ; 3189
                00000000G       EF      01  FB  00052             CALLS   #1, RELEASE_RECORD                      ;
                                        58  D5  00059 3$:          TSTL    SQX_N                                   ; 3190
                                        03  12  0005B             BNEQ    4$                                      ;
                                    00BC  31  0005D               BRW     13$                                     ;
                                        58  DD  00060 4$:          PUSHL   SQX_N                                   ; 3195
                00000000G       EF      01  FB  00062             CALLS   #1, READ_RECORD                         ;
                                    6E      50  DO  00069         MOVL    R0, SQX                                 ;
                    57              6E      0C  C1  0006C         ADDL3   #12, SQX, SQE                           ; 3200
                                            5A  D4  00070         CLRL    SQE_N                                   ; 3201
                                            67  95  00072 5$:     TSTB    (SQE)                                   ; 3203
                                            03  12  00074         BNEQ    6$                                      ;
                                        0090  31  00076           BRW     12$                                     ;
                        59          24  A7  DO  00079 6$:          MOVL    36(SQE), SMQ_N                          ; 3211
                                        59  DD  0007D             PUSHL   SMQ_N                                   ;
                00000000G       EF      01  FB  0007F             CALLS   #1, READ_RECORD                         ;
                                    54      50  DO  00086         MOVL    R0, SMQ                                 ;
                        56          2C  A4  D1  00089             CMPL    44(SMQ), R6                             ; 3217
                                        46  13  0008D             BEQL    8$                                      ;
                                    48  A4  DD  0008F             PUSHL   72(SMQ)                                 ; 3218
                                        56  DD  00092             PUSHL   R6                                      ;
                    FF1B            CF      02  FB  00094         CALLS   #2, FIND_QUEUE_REFERENCES_J             ;
                                    39      50  E8  00099         BLBS    R0, 8$                                  ;
                                    78  A4  DD  0009C             PUSHL   120(SMQ)                                ; 3219
                                        56  DD  0009F             PUSHL   R6                                      ;
                    FF0E            CF      02  FB  000A1         CALLS   #2, FIND_QUEUE_REFERENCES_J             ;
                                    2C      50  E8  000A6         BLBS    R0, 8$                                  ;
                                    74  A4  D5  000A9             TSTL    116(SMQ)                                ; 3230
                                        49  13  000AC             BEQL    11$                                     ;
                        55          74  A4  DO  000AE             MOVL    116(SMQ), AUX_N                         ; 3237
                                        55  DD  000B2             PUSHL   AUX_N                                   ;
                00000000G       EF      01  FB  000B4             CALLS   #1, READ_RECORD                         ;
                                    53      50  DO  000BB         MOVL    R0, AUX                                 ;
                52          0C  A3          01  C1  000BE         ADDL3   #1, 12(AUX), N                          ; 3238
                                            26  11  000C3         BRB     10$                                     ;
                        56          0C A342  D1  000C5 7$:          CMPL    12(AUX)[N], R6                          ; 3240
                                            1F  12  000CA         BNEQ    10$                                     ;
                                        55  DD  000CC             PUSHL   AUX_N                                   ; 3243
                00000000G       EF      01  FB  000CE             CALLS   #1, RELEASE_RECORD                      ;
                                        59  DD  000D5 8$:          PUSHL   SMQ_N                                   ; 3244
                00000000G       EF      01  FB  000D7             CALLS   #1, RELEASE_RECORD                      ;
                                        58  DD  000DE             PUSHL   SQX_N                                   ; 3245
                00000000G       EF      01  FB  000E0 9$:          CALLS   #1, RELEASE_RECORD                      ;
                                    50      01  DO  000E7         MOVL    #1, R0                                  ; 3246
                                            04  000EA             RET                                             ;
                        D7              52  F5  000EB 10$:          SOBGTR  N, 7$                                   ; 3238
                                        55  DD  000EE             PUSHL   AUX_N                                   ; 3249
                00000000G       EF      01  FB  000F0             CALLS   #1, RELEASE_RECORD                      ;
                                        59  DD  000F7 11$:          PUSHL   SMQ_N                                   ; 3255
                00000000G       EF      01  FB  000F9             CALLS   #1, RELEASE_RECORD                      ;
                                    57      28  C0  00100         ADDL2   #40, SQE                                ; 3259
    FF69            5A              01      08  F1  00103         ACBL    #11, #1, SQE_N, 5$                      ; 3201
                                    5B      00  BE  D0  00109 12$:  MOVL    @SQX, SQX_NS                            ; 3265
```

```
                                    58  DD 0010D          PUSHL   SQX_N                                   ; 3266
                                    01  FB 0010F          CALLS   #1,_RELEASE_RECORD                      ;
                        00000000G   EF                                                                    ;
                                    5B  D0 00116          MOVL    SQX_NS, SQX_N                           ; 3267
                                    58                                                                    ;
                                FF3D  31 00119          BRW     3$                                      ; 3190
                                    50  D4 0011C 13$:     CLRL    R0                                      ; 3272
                                        04 0011E          RET                                             ;

; Routine Size:  287 bytes,    Routine Base:  CODE + 0C94
```

```
;  2252     3273  1 GLOBAL ROUTINE FIND_QUEUE(DESC;SQX_N,SQE,SMQ_N,SMQ): L_OUTPUT_4=
;  2253     3274  1
;  2254     3275  1 !++
;  2255     3276  1 !
;  2256     3277  1 ! FUNCTIONAL DESCRIPTION:
;  2257     3278  1 !       This routine finds a queue header.
;  2258     3279  1 !
;  2259     3280  1 ! INPUT PARAMETERS:
;  2260     3281  1 !       DESC            - Short descriptor for queue name.
;  2261     3282  1 !
;  2262     3283  1 ! IMPLICIT INPUTS:
;  2263     3284  1 !       NONE
;  2264     3285  1 !
;  2265     3286  1 ! OUTPUT PARAMETERS:
;  2266     3287  1 !       SQX_N           - Record number of SQX.
;  2267     3288  1 !       SQE             - Pointer to SQX entry.
;  2268     3289  1 !       SMQ_N           - Record number of SMQ.
;  2269     3290  1 !       SMQ             - Pointer to SMQ.
;  2270     3291  1 !
;  2271     3292  1 ! IMPLICIT OUTPUTS:
;  2272     3293  1 !       NONE
;  2273     3294  1 !
;  2274     3295  1 ! ROUTINE VALUE:
;  2275     3296  1 !       True if the queue exists, false otherwise.
;  2276     3297  1 !
;  2277     3298  1 ! SIDE EFFECTS:
;  2278     3299  1 !       NONE
;  2279     3300  1 !
;  2280     3301  1 !--
;  2281     3302  1
;  2282     3303  2 BEGIN
;  2283     3304  2 MAP
;  2284     3305  2         DESC:           REF BBLOCK,     ! Short descriptor for queue name
;  2285     3306  2         SQE:            REF BBLOCK,     ! Pointer to SQX entry
;  2286     3307  2         SMQ:            REF BBLOCK;     ! Pointer to SMQ
;  2287     3308  2 LOCAL
;  2288     3309  2         SQX:            REF BBLOCK,     ! Pointer to SQX
;  2289     3310  2         SQX_NS;                         ! Record number of successor of SQX
;  2290     3311  2
;  2291     3312  2
;  2292     3313  2 ! Search the queue index for the desired name.
;  2293     3314  2 !
;  2294     3315  2 SQX = READ_RECORD(SQH$K_RECNO);
;  2295     3316  2 SQX_N = .SQX[SQH$L_QUEUE_INDEX_LIST];
;  2296     3317  2 RELEASE_RECORD(SQH$K_RECNO);
;  2297     3318  2 WHILE .SQX_N NEQ 0 DO
;  2298     3319  2     BEGIN
;  2299     3320  2
;  2300     3321  2     ! Read the queue index record.
;  2301     3322  2     !
;  2302     3323  2     SQX = READ_RECORD(.SQX_N);
;  2303     3324  2
;  2304     3325  2
;  2305     3326  2     ! Search the queue index for the desired name.
;  2306     3327  2     !
;  2307     3328  3     SQE = SQX[SYM$T_DATA];
;  2308     3329  3     INCR SQE_N FROM 0 TO SQX$K_ENTRIES-1 DO
```

```
;  2309          3330  4          BEGIN
;  2310          3331  4          IF CH$RCHAR(SQE[SQX$T_NAME]) EQL 0
;  2311          3332  4          THEN
;  2312          3333  4              EXITLOOP
;  2313          3334  4          ELSE
;  2314          3335  5              BEGIN
;  2315          3336  5              CASE CH$COMPARE(
;  2316          3337  5                  CH$RCHAR(SQE[SQX$T_NAME]), SQE[SQX$T_NAME]+1,
;  2317          3338  5                  .DESC[SDSC_W_LENGTH], .DESC[SDSC_A_POINTER],
;  2318          3339  5                  %C' ')
;  2319          3340  5              FROM -1 TO 1 OF
;  2320          3341  5                  SET
;  2321          3342  5
;  2322          3343  5                  [-1]:
;  2323          3344  5                      SQE = .SQE + SQX$S_SQX;
;  2324          3345  5
;  2325          3346  5                  [0]:
;  2326          3347  6                      BEGIN
;  2327          3348  6                      SMQ = READ_RECORD(SMQ_N = .SQE[SQX$L_QUEUE_LINK]);
;  2328          3349  6                      RETURN TRUE;
;  2329          3350  5                      END;
;  2330          3351  5
;  2331          3352  5                  [+1]:
;  2332          3353  6                      BEGIN
;  2333          3354  6                      RELEASE_RECORD(.SQX_N);
;  2334          3355  6                      RETURN FALSE;
;  2335          3356  5                      END;
;  2336          3357  5
;  2337          3358  5                  TES;
;  2338          3359  4              END;
;  2339          3360  3          END;
;  2340          3361  3
;  2341          3362  3
;  2342          3363  3          ! Advance to the next index block.
;  2343          3364  3          !
;  2344          3365  3          SQX_NS = .SQX[SYM$L_LINK];
;  2345          3366  3          RELEASE_RECORD(.SQX_N);
;  2346          3367  3          SQX_N = .SQX_NS;
;  2347          3368  2          END;
;  2348          3369  2
;  2349          3370  2
;  2350          3371  2      FALSE
;  2351          3372  1      END;


                                                00FC 00000          .ENTRY  FIND_QUEUE, Save R2,R3,R4,R5,R6,R7          ; 3273
                                    54 00000000G    EF   9E 00002   MOVAB   RELEASE_RECORD, R4
                                                    01   DD 00009   PUSHL   #1                                          ; 3315
                                00000000G       EF      01   FB 0000B   CALLS   #1, READ_RECORD
                                                55      50   D0 00012   MOVL    R0, SQX
                                                58  64  A5   D0 00015   MOVL    100(SQX), SQX_N                         ; 3316
                                                    01   DD 00019   PUSHL   #1                                          ; 3317
                                                64      01   FB 0001B   CALLS   #1, RELEASE_RECORD
                                                        58   D5 0001E 1$:   TSTL    SQX_N                               ; 3318
```

```
                                        5B  13  00020          BEQL    7$                                          ;
                                        58  DD  00022          PUSHL   SQX_N                                       ; 3323
                    00000000G   EF      01  FB  00024          CALLS   #1, READ_RECORD                             ;
                                55      50  D0  00027          MOVL    R0, SQX                                     ;
                                59      5B  0C  A5  9E  0002E  MOVAB   12(R5), SQE                                 ; 3328
                                        5B  D4  00032          CLRL    SQE_N                                       ; 3329
                                        69  95  00034  2$:     TSTB    (SQE)                                       ; 3331
                                        38  13  00036          BEQL    6$                                          ;
                                51      69  9A  00038          MOVZBL  (SQE), R1                                   ; 3337
                                50      04  AC  D0  0003B      MOVL    DESC, R0                                    ; 3338
                60      20      01      A9  51  2D  0003F      CMPC5   R1, 1(SQE), #32, (R0), @2(R0)               ; 3337
                                            02  B0  00045
                                            1C  1A  00047      BGTRU   4$                                          ;
                                            05  1E  00049      BGEQU   3$                                          ;
                                59          28  C0  0004B      ADDL2   #40, SQE                                    ; 3344
                                            1C  11  0004E      BRB     5$                                          ;
                                5A      24  A9  D0  00050  3$:  MOVL    36(SQE), SMQ_N                              ; 3348
                                            5A  DD  00054      PUSHL   SMQ_N                                       ;
                    00000000G   EF          01  FB  00056      CALLS   #1, READ_RECORD                             ;
                                57          50  D0  0005D      MOVL    R0, SMQ                                     ;
                                50          01  D0  00060      MOVL    #1, R0                                      ; 3349
                                            1A  11  00063      BRB     8$                                          ;
                                            58  DD  00065  4$:  PUSHL   SQX_N                                       ; 3354
                                64          01  FB  00067      CALLS   #1, RELEASE_RECORD                          ;
                                            11  11  0006A      BRB     7$                                          ; 3355
                        C4      5B          08  F3  0006C  5$:  AOBLEQ  #11, SQE_N, 2$                             ; 3329
                                56          65  D0  00070  6$:  MOVL    (SQX), SQX_NS                               ; 3365
                                            58  DD  00073      PUSHL   SQX_N                                       ; 3366
                                64          01  FB  00075      CALLS   #1, RELEASE_RECORD                          ;
                                58          56  D0  00078      MOVL    SQX_NS, SQX_N                               ; 3367
                                            A1  11  0007B      BRB     1$                                          ; 3318
                                            50  D4  0007D  7$:  CLRL    R0                                          ; 3372
                                5B          57  D0  0007F  8$:  MOVL    R7, R11                                     ;
                                            04  00082          RET                                                 ;

; Routine Size:  131 bytes,    Routine Base:  CODE + 0DB3
```

```
;  2353     3373  1 GLOBAL ROUTINE DEALLOCATE_VARIABLE_DATA(FIELD_SIZE,FIELD_ADDRESS): NOVALUE=
;  2354     3374  1
;  2355     3375  1 !++
;  2356     3376  1 !
;  2357     3377  1 ! FUNCTIONAL DESCRIPTION:
;  2358     3378  1 !       This routine deallocates extension records linked to a fixed/variable
;  2359     3379  1 !       data field, if they exist.
;  2360     3380  1 !
;  2361     3381  1 ! INPUT PARAMETERS:
;  2362     3382  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2363     3383  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2364     3384  1 !
;  2365     3385  1 ! IMPLICIT INPUTS:
;  2366     3386  1 !       NONE
;  2367     3387  1 !
;  2368     3388  1 ! OUTPUT PARAMETERS:
;  2369     3389  1 !       NONE
;  2370     3390  1 !
;  2371     3391  1 ! IMPLICIT OUTPUTS:
;  2372     3392  1 !       NONE
;  2373     3393  1 !
;  2374     3394  1 ! ROUTINE VALUE:
;  2375     3395  1 !       NONE
;  2376     3396  1 !
;  2377     3397  1 ! SIDE EFFECTS:
;  2378     3398  1 !       NONE
;  2379     3399  1 !
;  2380     3400  1 !--
;  2381     3401  1
;  2382     3402  2 BEGIN
;  2383     3403  2 MAP
;  2384     3404  2         FIELD_ADDRESS:  REF BBLOCK;     ! Pointer to fixed/variable buffer
;  2385     3405  2
;  2386     3406  2
;  2387     3407  2 IF .FIELD_ADDRESS[FVDF_LENGTH] GTRU .FIELD_SIZE - 2
;  2388     3408  2 THEN
;  2389     3409  2     DEALLOCATE_RECORD_LIST(.FIELD_ADDRESS[FVDF_LINK]);
;  2390     3410  2
;  2391     3411  2
;  2392     3412  2 CH$FILL(0, .FIELD_SIZE, .FIELD_ADDRESS);
;  2393     3413  1 END;


                                                  003C 00000         .ENTRY  DEALLOCATE_VARIABLE_DATA, Save R2,R3,R4,R5 ; 3373
                                        52      08  AC  D0 00002         MOVL    FIELD_ADDRESS, R2                          ; 3407
                                 50     04  AC      02  C3 00006         SUBL3   #2, FIELD_SIZE, R0                         ;
                        50       62          10      00  ED 0000B         CMPZV   #0, #16, (R2), R0                          ;
                                                    0A  1B 00010         BLEQU   1$                                         ;
                                                  02    A2  DD 00012         PUSHL   2(R2)                                      ; 3409
                                    00000000G     EF      01  FB 00015         CALLS   #1, DEALLOCATE_RECORD_LIST                 ;
                04  AC          00          6E          00  2C 0001C 1$:     MOVC5   #0, (SP), #0, FIELD_SIZE, (R2)             ; 3412
                                                            62      00022                                                      ;
                                                              04 00023         RET                                            ; 3413
```

; Routine Size: 36 bytes,   Routine Base: CODE + 0E36

```
;  2395     3414  1 GLOBAL ROUTINE FETCH_VARIABLE_ITEM(FIELD_SIZE,FIELD_ADDRESS,ITEM_CODE,ITEM_BUFFER)=
;  2396     3415  1
;  2397     3416  1 !++
;  2398     3417  1 !
;  2399     3418  1 ! FUNCTIONAL DESCRIPTION:
;  2400     3419  1 !       This routine fetches an item from a fixed/variable data field.
;  2401     3420  1 !
;  2402     3421  1 ! INPUT PARAMETERS:
;  2403     3422  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2404     3423  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2405     3424  1 !       ITEM_CODE       - Type code for the item.
;  2406     3425  1 !       ITEM_BUFFER     - Pointer to the item buffer.
;  2407     3426  1 !
;  2408     3427  1 ! IMPLICIT INPUTS:
;  2409     3428  1 !       NONE
;  2410     3429  1 !
;  2411     3430  1 ! OUTPUT PARAMETERS:
;  2412     3431  1 !       NONE
;  2413     3432  1 !
;  2414     3433  1 ! IMPLICIT OUTPUTS:
;  2415     3434  1 !       NONE
;  2416     3435  1 !
;  2417     3436  1 ! ROUTINE VALUE:
;  2418     3437  1 !       Updated pointer to the item buffer.
;  2419     3438  1 !
;  2420     3439  1 ! SIDE EFFECTS:
;  2421     3440  1 !       NONE
;  2422     3441  1 !
;  2423     3442  1 !--
;  2424     3443  1
;  2425     3444  2 BEGIN
;  2426     3445  2 MAP
;  2427     3446  2         FIELD_ADDRESS:  REF BBLOCK;     ! Pointer to fixed/variable buffer
;  2428     3447  2 LOCAL
;  2429     3448  2         ITEM:           REF BBLOCK;     ! Cursor for item buffer
;  2430     3449  2
;  2431     3450  2
;  2432     3451  2 ITEM = .ITEM_BUFFER;
;  2433     3452  2 IF .FIELD_ADDRESS[FVDF_LENGTH] NEQ 0
;  2434     3453  2 THEN
;  2435     3454  3     BEGIN
;  2436     3455  3     ITEM[0,0,16,0] = .FIELD_ADDRESS[FVDF_LENGTH];
;  2437     3456  3     ITEM[2,0,16,0] = .ITEM_CODE;
;  2438     3457  3     ITEM = .ITEM + 4;
;  2439     3458  3     IF .FIELD_ADDRESS[FVDF_LENGTH] LEQU .FIELD_SIZE - 2
;  2440     3459  3     THEN
;  2441     3460  4         BEGIN
;  2442     3461  4         MOVC3(
;  2443     3462  4             FIELD_ADDRESS[FVDF_LENGTH],
;  2444     3463  4             FIELD_ADDRESS[FVDF_DATA],
;  2445     3464  4             .ITEM; ... ITEM);
;  2446     3465  4         END
;  2447     3466  3     ELSE
;  2448     3467  4         BEGIN
;  2449     3468  4         LOCAL
;  2450     3469  4             AUX_N,                     ! Record number of auxiliary record
;  2451     3470  4             INPUT_LENGTH;              ! Remaining input data
```

```
;  2452          3471  4
;  2453          3472  4
;  2454          3473  4          ! Initialize.
;  2455          3474  4          !
;  2456          3475  4          INPUT_LENGTH = .FIELD_ADDRESS[FVDF_LENGTH];
;  2457          3476  4          AUX_N = .FIELD_ADDRESS[FVDF_LINK];
;  2458          3477  4
;  2459          3478  4          ! Loop over all auxiliary information records.
;  2460          3479  4          !
;  2461          3480  4          WHILE .AUX_N NEQ 0 DO
;  2462          3481  4              BEGIN
;  2463          3482  5              LOCAL
;  2464          3483  5                  AUX_NS,                 ! Record number of successor of AUX
;  2465          3484  5                  AUX:    REF BBLOCK,     ! Pointer to auxiliary record
;  2466          3485  5                  THIS_LENGTH;            ! Length of current transfer
;  2467          3486  5
;  2468          3487  5
;  2470          3489  5                  AUX = READ_RECORD(.AUX_N);
;  2471          3490  5                  THIS_LENGTH = MINU(.INPUT_LENGTH, SYM$S_DATA);
;  2472          3491  5                  MOVC3(INPUT_LENGTH, AUX[SYM$T_DATA], .ITEM; ..., ITEM);
;  2473          3492  5                  INPUT_LENGTH = .INPUT_LENGTH - .THIS_LENGTH;
;  2474          3493  5                  AUX_NS = .AUX[SYM$L_LINK];
;  2475          3494  5                  RELEASE_RECORD(.AUX_N);
;  2476          3495  5                  AUX_N = .AUX_NS;
;  2477          3496  4                  END;
;  2478          3497  3              END;
;  2479          3498  2          END;
;  2480          3499  2      .ITEM
;  2481          3500  1  END;


                                        03FC 00000          .ENTRY  FETCH_VARIABLE_ITEM, Save R2,R3,R4,R5,R6,-      ; 3414
                                                                    R7,R8,R9
                                53      10   AC  D0 00002          MOVL    ITEM_BUFFER, ITEM                       ; 3451
                                57      08   AC  D0 00006          MOVL    FIELD_ADDRESS, R7                       ; 3452
                                             67  B5 0000A          TSTW    (R7)                                    ;
                                             5C  13 0000C          BEQL    4$                                      ;
                                83           67  B0 0000E          MOVW    (R7), (ITEM)+                           ; 3455
                                83      0C   AC  B0 00011          MOVW    ITEM_CODE, (ITEM)+                      ; 3456
                          50        04   AC  02  C3 00015          SUBL3   #2, FIELD_SIZE, R0                      ; 3458
                50        67            10  00  ED 0001A          CMPZV   #0, #16, (R7), R0                       ;
                                             07  1A 0001F          BGTRU   1$                                      ;
                          63        02   A7  67  28 00021          MOVC3   (R7), 2(R7), (ITEM)                     ; 3464
                                             42  11 00026          BRB     4$                                      ; 3458
                                56           67  3C 00028  1$:     MOVZWL  (R7), INPUT_LENGTH                      ; 3475
                                58      02   A7  D0 0002B          MOVL    2(R7), AUX_N                            ; 3476
                                             39  13 0002F  2$:     BEQL    4$                                      ; 3481
                                             58  DD 00031          PUSHL   AUX_N                                   ; 3489
                     00000000G  EF           01  FB 00033          CALLS   #1, READ_RECORD                         ;
                                57           50  D0 0003A          MOVL    R0, AUX                                 ;
                                50           56  D0 0003D          MOVL    INPUT_LENGTH, R0                        ; 3490
                     000001F4   8F           50  D1 00040          CMPL    R0, #500                                ;
                                             05  1B 00047          BLEQU   3$                                      ;
```

```
                        50      01F4   8F  3C 00049          MOVZWL  #500, R0                                  ;
                        59             50  D0 0004E 38:      MOVL    R0, THIS_LENGTH                         ;
                63      0C      A7     56  28 00051          MOVC3   INPUT_LENGTH, 12(AUX), (ITEM)           ; 3491
                                56     59  C2 00056          SUBL2   THIS_LENGTH, INPUT_LENGTH               ; 3492
                                52     67  D0 00059          MOVL    (AUX), AUX_NS                           ; 3493
                                       58  DD 0005C          PUSHL   AUX_N                                   ; 3494
                00000000G       EF     01  FB 0005E          CALLS   #1, RELEASE_RECORD                      ;
                                58     52  D0 00065          MOVL    AUX_NS, AUX_N                           ; 3495
                                       C5  11 00068          BRB     2$                                      ; 3481
                                50     53  D0 0006A 48:      MOVL    ITEM, R0                                ; 3500
                                           04 0006D          RET                                             ;

; Routine Size:  110 bytes,    Routine Base:  CODE + 0E5A
```

```
;  2483          3501  1 GLOBAL ROUTINE FETCH_VARIABLE_ITEM_LIST(FIELD_SIZE,FIELD_ADDRESS,ITEM_CODE,ITEM_BUFFER)=
;  2484          3502  1
;  2485          3503  1 !++
;  2486          3504  1 !
;  2487          3505  1 ! FUNCTIONAL DESCRIPTION:
;  2488          3506  1 !       This routine fetches a sequence of items from a fixed/variable data
;  2489          3507  1 !       field.
;  2490          3508  1 !
;  2491          3509  1 ! INPUT PARAMETERS:
;  2492          3510  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2493          3511  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2494          3512  1 !       ITEM_CODE       - Type code for the first item, incrementing by 1.
;  2495          3513  1 !       ITEM_BUFFER     - Pointer to the item buffer.
;  2496          3514  1 !
;  2497          3515  1 ! IMPLICIT INPUTS:
;  2498          3516  1 !       NONE
;  2499          3517  1 !
;  2500          3518  1 ! OUTPUT PARAMETERS:
;  2501          3519  1 !       NONE
;  2502          3520  1 !
;  2503          3521  1 ! IMPLICIT OUTPUTS:
;  2504          3522  1 !       NONE
;  2505          3523  1 !
;  2506          3524  1 ! ROUTINE VALUE:
;  2507          3525  1 !       Updated pointer to the item buffer.
;  2508          3526  1 !
;  2509          3527  1 ! SIDE EFFECTS:
;  2510          3528  1 !       NONE
;  2511          3529  1 !
;  2512          3530  1 !--
;  2513          3531  1
;  2514          3532  2 BEGIN
;  2515          3533  2 MAP
;  2516          3534  2         FIELD_ADDRESS:  REF BBLOCK;
;  2517          3535  2 LOCAL
;  2518          3536  2         ITEM:           REF BBLOCK,
;  2519          3537  2         A:              REF BBLOCK,
;  2520          3538  2         E,
;  2521          3539  2         I,
;  2522          3540  2         BUFFER:         BBLOCK[1024];
;  2523          3541  2
;  2524          3542  2
;  2525          3543  2 ITEM = .ITEM_BUFFER;
;  2526          3544  2 FETCH_VARIABLE_DATA(.FIELD_SIZE, .FIELD_ADDRESS, 1024, BUFFER);
;  2527          3545  2 A = BUFFER;
;  2528          3546  2 E = .A + .FIELD_ADDRESS[FVDF_LENGTH];
;  2529          3547  2 I = .ITEM_CODE;
;  2530          3548  2 WHILE .A [SSA .E DO
;  2531          3549  3     BEGIN
;  2532          3550  3     LOCAL
;  2533          3551  3         L;
;  2534          3552  3
;  2535          3553  3     L = .A[0,0,16,0];
;  2536          3554  3     A = .A + 2;
;  2537          3555  3     IF .L NEQ 0
;  2538          3556  3     THEN
;  2539          3557  4         BEGIN
```

```
;  2540          3558  4          ITEM[0,0,16,0] = .L;
;  2541          3559  4          ITEM[2,0,16,0] = .I;
;  2542          3560  4          ITEM = .ITEM + 4;
;  2543          3561  4          MOVC3(L, .A, .ITEM); , A, , ITEM);
;  2544          3562  3          END;
;  2545          3563  3      I = .I + 1;
;  2546          3564  2      END;
;  2547          3565  2 .ITEM
;  2548          3566  1 END;


                                        00FC 00000          .ENTRY  FETCH_VARIABLE_ITEM_LIST, Save R2,R3,R4,R5,-  3501
                                                                    R6,R7
                                5E    FC00   CE  9E 00002           MOVAB   -1024(SP), SP
                                53      10   AC  D0 00007           MOVL    ITEM_BUFFER, ITEM                          3543
                                             5E  DD 0000B           PUSHL   SP                                         3544
                                7E    0400   8F  3C 0000D           MOVZWL  #1024, -(SP)
                                7E      04   AC  7D 00012           MOVQ    FIELD_SIZE, -(SP)
                          0000V CF          04  FB 00016           CALLS   #4, FETCH_VARIABLE_DATA
                                51          6E  9E 0001B           MOVAB   BUFFER, A                                  3545
                                57      08  BC  3C 0001E           MOVZWL  @FIELD_ADDRESS, E                          3546
                                57          51  C0 00022           ADDL2   A, E
                                56      0C  AC  D0 00025           MOVL    ITEM_CODE, I                               3547
                                57          51  D1 00029 1$:       CMPL    A, E                                       3548
                                            13  1E 0002C           BGEQU   3$
                                50          81  3C 0002E           MOVZWL  (A)+, L                                    3553
                                            0A  13 00031           BEQL    2$                                         3555
                                83          50  B0 00033           MOVW    L, (ITEM)+                                 3558
                                83          56  B0 00036           MOVW    I, (ITEM)+                                 3559
                          63            61      50  28 00039       MOVC3   L, (A), (ITEM)                             3561
                                            56  D6 0003D 2$:       INCL    I                                          3563
                                            E8  11 0003F           BRB     1$                                         3548
                                50          53  D0 00041 3$:       MOVL    ITEM, R0                                   3566
                                            04 00044               RET

; Routine Size:  69 bytes,    Routine Base:  CODE + 0EC8
```

```
;  2550     3567  1 GLOBAL ROUTINE FETCH_VARIABLE_DATA(FIELD_SIZE,FIELD_ADDRESS,BUFFER_LENGTH,BUFFER_ADDRESS): NOVALUE=
;  2551     3568  1
;  2552     3569  1 !++
;  2553     3570  1 !
;  2554     3571  1 ! FUNCTIONAL DESCRIPTION:
;  2555     3572  1 !       This routine fetches data from a fixed/variable data field.
;  2556     3573  1 !
;  2557     3574  1 ! INPUT PARAMETERS:
;  2558     3575  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2559     3576  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2560     3577  1 !       BUFFER_LENGTH   - Descriptor for output buffer.
;  2561     3578  1 !       BUFFER_ADDRESS  -
;  2562     3579  1 !
;  2563     3580  1 ! IMPLICIT INPUTS:
;  2564     3581  1 !       NONE
;  2565     3582  1 !
;  2566     3583  1 ! OUTPUT PARAMETERS:
;  2567     3584  1 !       NONE
;  2568     3585  1 !
;  2569     3586  1 ! IMPLICIT OUTPUTS:
;  2570     3587  1 !       NONE
;  2571     3588  1 !
;  2572     3589  1 ! ROUTINE VALUE:
;  2573     3590  1 !       NONE
;  2574     3591  1 !
;  2575     3592  1 ! SIDE EFFECTS:
;  2576     3593  1 !       NONE
;  2577     3594  1 !
;  2578     3595  1 !--
;  2579     3596  1
;  2580     3597  2 BEGIN
;  2581     3598  2 MAP
;  2582     3599  2         FIELD_ADDRESS:  REF BBLOCK;     ! Pointer to fixed/variable buffer
;  2583     3600  2 LOCAL
;  2584     3601  2         AUX_N,                          ! Record number of auxiliary record
;  2585     3602  2         AUX_NS,                         ! Record number of successor of AUX
;  2586     3603  2         AUX:            REF BBLOCK;     ! Pointer to auxiliary record
;  2587     3604  2
;  2588     3605  2
;  2589     3606  2 IF .FIELD_ADDRESS[FVDF_LENGTH] LEQU .FIELD_SIZE - 2
;  2590     3607  2 THEN
;  2591     3608  3     BEGIN
;  2592     3609  3     CH$COPY(
;  2593     3610  3         .FIELD_ADDRESS[FVDF_LENGTH], FIELD_ADDRESS[FVDF_DATA],
;  2594     3611  3         0,
;  2595     3612  3         .BUFFER_LENGTH, .BUFFER_ADDRESS);
;  2596     3613  3     END
;  2597     3614  2 ELSE
;  2598     3615  3     BEGIN
;  2599     3616  3     LOCAL
;  2600     3617  3         INPUT_LENGTH,                   ! Remaining input data
;  2601     3618  3         CURRENT_LENGTH,                 ! Remaining buffer length
;  2602     3619  3         CURRENT_ADDRESS;                ! Current buffer address
;  2603     3620  3
;  2604     3621  3
;  2605     3622  3     ! Initialize.
;  2606     3623  3     !
```

```
;  2607          3624  3      INPUT_LENGTH = .FIELD_ADDRESS[FVDF_LENGTH];
;  2608          3625  3      CURRENT_LENGTH = .BUFFER_LENGTH;
;  2609          3626  3      CURRENT_ADDRESS = .BUFFER_ADDRESS;
;  2610          3627  3      AUX_N = .FIELD_ADDRESS[FVDF_LINK];
;  2611          3628  3
;  2612          3629  3
;  2613          3630  3      ! Loop over all auxiliary information records.
;  2614          3631  3      !
;  2615          3632  3      WHILE TRUE DO
;  2616          3633  4          BEGIN
;  2617          3634  4          AUX = READ_RECORD(.AUX_N);
;  2618          3635  4          IF .AUX[SYM$L_LINK] EQL 0
;  2619          3636  4          THEN
;  2620          3637  5              BEGIN
;  2621          3638  5              CH$COPY(
;  2622          3639  5                  MINU(.INPUT_LENGTH, SYM$S_DATA), AUX[SYM$T_DATA],
;  2623          3640  5                  0,
;  2624          3641  5                  .CURRENT_LENGTH, .CURRENT_ADDRESS);
;  2625          3642  5              RELEASE_RECORD(.AUX_N);
;  2626          3643  5              EXITLOOP;
;  2627          3644  5              END
;  2628          3645  4          ELSE
;  2629          3646  5              BEGIN
;  2630          3647  5              LOCAL
;  2631          3648  5                  THIS_LENGTH;            ! Length of current transfer
;  2632          3649  5
;  2633          3650  5
;  2634          3651  5              THIS_LENGTH = MINU(.CURRENT_LENGTH, SYM$S_DATA);
;  2635          3652  5              CURRENT_ADDRESS = CH$MOVE(
;  2636          3653  5                  .THIS_LENGTH, AUX[SYM$T_DATA], .CURRENT_ADDRESS);
;  2637          3654  5              CURRENT_LENGTH = .CURRENT_LENGTH - .THIS_LENGTH;
;  2638          3655  5              IF .CURRENT_LENGTH EQL 0 THEN EXITLOOP;
;  2639          3656  5              AUX_NS = .AUX[SYM$L_LINK];
;  2640          3657  5              RELEASE_RECORD(.AUX_N);
;  2641          3658  5              AUX_N = .AUX_NS;
;  2642          3659  4              END;
;  2643          3660  3          END;
;  2644          3661  2      END;
;  2645          3662  1  END;


                                                    OFFC 00000          .ENTRY  FETCH_VARIABLE_DATA, Save R2,R3,R4,R5,R6,-      ; 3567
                                                                                R7,R8,R9,R10,R11
                                                5E          04  C2 00002        SUBL2   #4, SP                                  ;
                                                58      08  AC  D0 00005        MOVL    FIELD_ADDRESS, R8                       ; 3610
                                        50      04  AC      02  C3 00009        SUBL3   #2, FIELD_SIZE, R0                      ; 3606
                                50      08  BC              00  ED 0000E        CMPZV   #0, #16, @FIELD_ADDRESS, R0             ;
                                                            0B  1A 00014        BGTRU   1$                                      ;
                            0C  AC          00      02  A8  08  BC  2C 00016    MOVC5   @FIELD_ADDRESS, 2(R8), #0, BUFFER_LENGTH, - ; 3612
                                                            10  BC      0001E           @BUFFER_ADDRESS                         ;
                                                                04  00020        RET                                             ; 3606
                                                57      08  BC  3C 00021 1$:   MOVZWL  @FIELD_ADDRESS, INPUT_LENGTH           ; 3624
                                                56      0C  AC  D0 00025        MOVL    BUFFER_LENGTH, CURRENT_LENGTH           ; 3625
                                                5B      10  AC  D0 00029        MOVL    BUFFER_ADDRESS, CURRENT_ADDRESS         ; 3626
```

```
                                5A      02  A8  D0 0002D          MOVL    2(R8), AUX_N                        ; 3627
                                            5A  DD 00031 2$:     PUSHL   AUX_N                               ; 3634
                00000000G       EF          01  FB 00033         CALLS   #1, READ_RECORD
                                59          50  D0 0003A         MOVL    R0, AUX
                                            69  D5 0003D         TSTL    (AUX)                               ; 3635
                                            22  12 0003F         BNEQ    4$
                                50          57  D0 00041         MOVL    INPUT_LENGTH, R0                    ; 3639
                000001F4        8F          50  D1 00044         CMPL    R0, #500
                                            05  1B 0004B         BLEQU   3$
                                50      01F4    8F  3C 0004D     MOVZWL  #500, R0
          56            00          0C    A9  50  2C 00052 3$:   MOVC5   R0, 12(AUX), #0, CURRENT_LENGTH, -  ; 3641
                                                6B      00058           (CURRENT_ADDRESS)
                                            5A  DD 00059         PUSHL   AUX_N                               ; 3642
                00000000G       EF          01  FB 0005B         CALLS   #1, RELEASE_RECORD
                                            04  00062         RET                                             ; 3637
                                50          56  D0 00063 4$:     MOVL    CURRENT_LENGTH, R0                  ; 3651
                000001F4        8F          50  D1 00066         CMPL    R0, #500
                                            05  1B 0006D         BLEQU   5$
                                50      01F4    8F  3C 0006F     MOVZWL  #500, R0
                                58          50  D0 00074 5$:     MOVL    R0, THIS_LENGTH
                    6B          0C    A9      58  28 00077       MOVC3   THIS_LENGTH, 12(AUX), (CURRENT_ADDRESS)  ; 3653
                                5B          53  D0 0007C         MOVL    R3, CURRENT_ADDRESS
                                56          58  C2 0007F         SUBL2   THIS_LENGTH, CURRENT_LENGTH         ; 3654
                                            11  13 00082         BEQL    6$                                  ; 3655
                                6E          69  D0 00084         MOVL    (AUX), AUX_NS                       ; 3656
                                            5A  DD 00087         PUSHL   AUX_N                               ; 3657
                00000000G       EF          01  FB 00089         CALLS   #1, RELEASE_RECORD
                                5A          6E  D0 00090         MOVL    AUX_NS, AUX_N                       ; 3658
                                            9C  11 00093         BRB     2$                                  ; 3632
                                                04 00095 6$:     RET                                         ; 3662

; Routine Size:  150 bytes,    Routine Base:  CODE + 0F0D
```

```
;  2647          3663  1 GLOBAL ROUTINE STORE_VARIABLE_DATA(RECORD_ADDRESS,FIELD_SIZE,FIELD_ADDRESS,TYPE_CODE,DATA_LENGTH,DATA_ADDRES
;  2648          3664  1
;  2649          3665  1 !++
;  2650          3666  1 !
;  2651          3667  1 ! FUNCTIONAL DESCRIPTION:
;  2652          3668  1 !       This routine stores data in a fixed/variable data field.  These fields
;  2653          3669  1 !       allow a string up to 65535 bytes to be stored and retrieved by use of
;  2654          3670  1 !       extension queue records; however, a string that does not exceed the
;  2655          3671  1 !       fixed field size is stored without use of auxiliary records.
;  2656          3672  1 !
;  2657          3673  1 ! INPUT PARAMETERS:
;  2658          3674  1 !       RECORD_ADDRESS  - Pointer to record containing the fixed/variable data
;  2659          3675  1 !                         field.
;  2660          3676  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2661          3677  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2662          3678  1 !       TYPE_CODE       - Value of SYM$B_TYPE for extension records.
;  2663          3679  1 !       DATA_LENGTH     - Descriptor for data to be stored.
;  2664          3680  1 !       DATA_ADDRESS    -
;  2665          3681  1 !
;  2666          3682  1 ! IMPLICIT INPUTS:
;  2667          3683  1 !       NONE
;  2668          3684  1 !
;  2669          3685  1 ! OUTPUT PARAMETERS:
;  2670          3686  1 !       NONE
;  2671          3687  1 !
;  2672          3688  1 ! IMPLICIT OUTPUTS:
;  2673          3689  1 !       NONE
;  2674          3690  1 !
;  2675          3691  1 ! ROUTINE VALUE:
;  2676          3692  1 !       Completion status.
;  2677          3693  1 !
;  2678          3694  1 ! SIDE EFFECTS:
;  2679          3695  1 !       NONE
;  2680          3696  1 !
;  2681          3697  1 !--
;  2682          3698  1
;  2683          3699  2 BEGIN
;  2684          3700  2 MAP
;  2685          3701  2         RECORD_ADDRESS: REF BBLOCK,     ! Pointer to record
;  2686          3702  2         FIELD_ADDRESS:  REF BBLOCK;     ! Pointer to fixed/variable buffer
;  2687          3703  2
;  2688          3704  2
;  2689          3705  2 IF .FIELD_ADDRESS[FVDF_LENGTH] NEQ 0
;  2690          3706  2 THEN
;  2691          3707  2     DEALLOCATE_VARIABLE_DATA(.FIELD_SIZE, .FIELD_ADDRESS);
;  2692          3708  2
;  2693          3709  2
;  2694          3710  2 FIELD_ADDRESS[FVDF_LENGTH] = .DATA_LENGTH;
;  2695          3711  2 IF .DATA_LENGTH LEQU .FIELD_SIZE - 2
;  2696          3712  2 THEN
;  2697          3713  3     BEGIN
;  2698          3714  3     CH$COPY(
;  2699          3715  3         .DATA_LENGTH, .DATA_ADDRESS,
;  2700          3716  3         0,
;  2701          3717  3         .FIELD_SIZE-2, FIELD_ADDRESS[FVDF_DATA]);
;  2702          3718  3     END
;  2703          3719  2 ELSE
```

```
;  2704          3720  3     BEGIN
;  2705          3721  3     LOCAL
;  2706          3722  3         SEQUENCE,                       ! Sequence counter
;  2707          3723  3         AUX_NP,                         ! Record number of predecessor of AUX
;  2708          3724  3         AUX_P:          REF BBLOCK,     ! Pointer to predecessor of AUX
;  2709          3725  3         AUX_N,                          ! Record number of auxiliary record
;  2710          3726  3         AUX:            REF BBLOCK,     ! Pointer to auxiliary record
;  2711          3727  3         CURRENT_LENGTH,                 ! Remaining source length
;  2712          3728  3         CURRENT_ADDRESS;                ! Current source address
;  2713          3729  3
;  2714          3730  3
;  2715          3731  3         ! Initialize.
;  2716          3732  3         !
;  2717          3733  3         SEQUENCE = 0;
;  2718          3734  3         AUX_NP = 0;
;  2719          3735  3         CURRENT_LENGTH = .DATA_LENGTH;
;  2720          3736  3         CURRENT_ADDRESS = .DATA_ADDRESS;
;  2721          3737  3
;  2722          3738  3
;  2723          3739  3         ! Loop until all source data is stored.
;  2724          3740  3         !
;  2725          3741  3         WHILE TRUE DO
;  2726          3742  4             BEGIN
;  2727          3743  4             LOCAL
;  2728          3744  4                 THIS_LENGTH,            ! Current transfer length
;  2729          3745  4                 STATUS;                 ! Status return
;  2730          3746  4
;  2731          3747  4
;  2732          3748  4             ! Obtain the minimum of the remaining input length and the space
;  2733          3749  4             ! available in one record.
;  2734          3750  4             !
;  2735          3751  4             THIS_LENGTH = .CURRENT_LENGTH;
;  2736          3752  4             IF .THIS_LENGTH GTRU SYMSS_DATA THEN THIS_LENGTH = SYMSS_DATA;
;  2737          3753  4
;  2738          3754  4
;  2739          3755  4             ! Allocate the record and set up the forward link.
;  2740          3756  4             !
;  2741          3757  4             STATUS = ALLOCATE_RECORD( ; AUX_N, AUX);
;  2742          3758  4             IF NOT .STATUS
;  2743          3759  4             THEN
;  2744          3760  5                 BEGIN
;  2745          3761  5                 DEALLOCATE_RECORD_LIST(.FIELD_ADDRESS[FVDF_LINK]);
;  2746          3762  5                 CH$FILL(0, .FIELD_SIZE, .FIELD_ADDRESS);
;  2747          3763  5                 RETURN .STATUS;
;  2748          3764  4                 END;
;  2749          3765  4             IF .AUX_NP EQL 0
;  2750          3766  4             THEN
;  2751          3767  4                 FIELD_ADDRESS[FVDF_LINK] = .AUX_N
;  2752          3768  4             ELSE
;  2753          3769  5                 BEGIN
;  2754          3770  5                 AUX_P[SYM$L_LINK] = .AUX_N;
;  2755          3771  5                 REWRITE_RECORD(.AUX_NP);
;  2756          3772  4                 END;
;  2757          3773  4
;  2758          3774  4
;  2759          3775  4             ! Initialize the record header.
;  2760          3776  4             !
```

```
;  2761          3777  4          SEQUENCE = .SEQUENCE + 1;
;  2762          3778  4          AUX[SYMSB_TYPE] = .TYPE_CODE;
;  2763          3779  4          AUX[SYMSB_AUX_SEQUENCE] = .SEQUENCE;
;  2764          3780  4          AUX[SYMSW_SEQUENCE] = .RECORD_ADDRESS[SYMSW_SEQUENCE];
;  2765          3781  4          AUX[SYMSL_ENTRY_NUMBER] = .RECORD_ADDRESS[SYMSL_ENTRY_NUMBER];
;  2766          3782  4
;  2767          3783  4
;  2768          3784  4          ! Move the information.
;  2769          3785  4          !
;  2770          3786  4          CH$MOVE(.THIS_LENGTH, .CURRENT_ADDRESS, AUX[SYMST_DATA]);
;  2771          3787  4
;  2772          3788  4
;  2773          3789  4          ! Update current length and address for the next record and quit if
;  2774          3790  4          ! all data has been transferred.
;  2775          3791  4          !
;  2776          3792  4          CURRENT_LENGTH = .CURRENT_LENGTH - .THIS_LENGTH;
;  2777          3793  4          IF .CURRENT_LENGTH EQL 0 THEN EXITLOOP;
;  2778          3794  4          CURRENT_ADDRESS = .CURRENT_ADDRESS + .THIS_LENGTH;
;  2779          3795  3          END;
;  2780          3796  3
;  2781          3797  3
;  2782          3798  3      REWRITE_RECORD(.AUX_N);
;  2783          3799  2      END;
;  2784          3800  2
;  2785          3801  2
;  2786          3802  2 SS$_NORMAL
;  2787          3803  1 END;
;  INFO#250                  L1:3770
;  Referenced LOCAL symbol AUX_P is probably not initialized
```

```
                                    0FFC 00000          .ENTRY  STORE_VARIABLE_DATA, Save R2,R3,R4,R5,R6,-      ; 3663
                                                                R7,R8,R9,R10,R11
                                5E          18  C2 00002       SUBL2   #24, SP
                                56      0C  AC  D0 00005       MOVL    FIELD_ADDRESS, R6                       ; 3705
                                            66  B5 00009       TSTW    (R6)
                                            0A  13 0000B       BEQL    1$
                                            56  DD 0000D       PUSHL   R6                                      ; 3707
                                        08  AC  DD 0000F       PUSHL   FIELD_SIZE
                            FE7C        CF      02  FB 00012   CALLS   #2, DEALLOCATE_VARIABLE_DATA
                                66      14  AC  B0 00017 1$:   MOVW    DATA_LENGTH, (R6)                       ; 3710
                        50      08      AC      02  C3 0001B   SUBL3   #2, FIELD_SIZE, R0                      ; 3711
                                50      14  AC  D1 00020       CMPL    DATA_LENGTH, R0
                                            0C  1A 00024       BGTRU   2$
                50      00      18      BC  14  AC  2C 00026   MOVC5   DATA_LENGTH, @DATA_ADDRESS, #0, R0, 2(R6)  ; 3717
                                            02  A6      0002D
                                        0092    31  0002F       BRW     9$                                      ; 3711
                                            0C  AE  7C 00032 2$:   CLRQ    AUX_NP                              ; 3734
                                04      AE  14  AC  7D 00035       MOVQ    DATA_LENGTH, CURRENT_LENGTH         ; 3735
                                        6E  04  AC  D0 0003A       MOVL    RECORD_ADDRESS, (SP)                ; 3780
                                        59  04  AC  D0 0003E       MOVL    RECORD_ADDRESS, R9                  ; 3781
                                        58  04  AE  D0 00042 3$:   MOVL    CURRENT_LENGTH, THIS_LENGTH         ; 3751
                        000001F4        8F      58  D1 00046       CMPL    THIS_LENGTH, #500                   ; 3752
                                                05  1B 0004D       BLEQU   4$
```

```
                        58      01F4    8F  3C 0004F        MOVZWL  #500, THIS_LENGTH                                       ;
                00000000G       EF      00  FB 00054 4$:    CALLS   #0, ALLOCATE_RECORD                                     ; 3757
                                57      50  D0 00058        MOVL    R0, STATUS                                              ;
                                15      57  E8 0005E        BLBS    STATUS, 5$                                              ; 3758
                                        02  A6  DD 00061    PUSHL   2(R6)                                                   ; 3761
                00000000G       EF      01  FB 00064        CALLS   #1, DEALLOCATE_RECORD_LIST                              ;
    08  AC          00          6E      00  2C 00068        MOVC5   #0, (SP), #0, FIELD_SIZE, (R6)                          ; 3762
                                                66 00071                                                                    ;
                                50      57  D0 00072        MOVL    STATUS, R0                                              ; 3763
                                        04  00075        RET                                                             ;
                                        0C  AE  D5 00076 5$:    TSTL    AUX_NP                                          ; 3765
                                        06  12 00079        BNEQ    6$                                                      ;
                            02  A6          5A  D0 0007B        MOVL    AUX_N, 2(R6)                                    ; 3767
                                        0E  11 0007F        BRB     7$                                                      ;
                            14  BE          5A  D0 00081 6$:    MOVL    AUX_N, @AUX_P                                   ; 3770
                                        0C  AE  DD 00085    PUSHL   AUX_NP                                                  ; 3771
                00000000G       EF      01  FB 00088        CALLS   #1, REWRITE_RECORD                                      ;
                                        10  AE  D6 0008F 7$:    INCL    SEQUENCE                                        ; 3777
                            04  AB      10  AC 90 00092        MOVB    TYPE_CODE, 4(AUX)                               ; 3778
                            05  AB      10  AE 90 00097        MOVB    SEQUENCE, 5(AUX)                                ; 3779
                    50          6E      06  C1 0009C        ADDL3   #6, (SP), R0                                            ; 3780
                            06  AB      60  B0 000A0        MOVW    (R0), 6(AUX)                                            ;
                            08  AB      08  A9 D0 000A4        MOVL    8(R9), 8(AUX)                                   ; 3781
                OC  AB      08  BE      58  28 000A9        MOVC3   THIS_LENGTH, @CURRENT_ADDRESS, 12(AUX)                  ; 3786
                            04  AE      58  C2 000AF        SUBL2   THIS_LENGTH, CURRENT_LENGTH                             ; 3792
                                        06  13 000B3        BEQL    8$                                                      ; 3793
                            08  AE      58  C0 000B5        ADDL2   THIS_LENGTH, CURRENT_ADDRESS                            ; 3794
                                        87  11 000B9        BRB     3$                                                      ; 3741
                                        5A  DD 000BB 8$:    PUSHL   AUX_N                                                   ; 3798
                00000000G       EF      01  FB 000BD        CALLS   #1, REWRITE_RECORD                                      ;
                                50      01  D0 000C4 9$:    MOVL    #1, R0                                                  ; 3803
                                                04 000C7        RET                                                         ;

; Routine Size:  200 bytes,    Routine Base:  CODE + 0FA3
```

```
;  2789     3804  1 GLOBAL ROUTINE STORE_VARIABLE_DATA_LIST(RECORD_ADDRESS,FIELD_SIZE,FIELD_ADDRESS,TYPE_CODE)=
;  2790     3805  1
;  2791     3806  1 !++
;  2792     3807  1 !
;  2793     3808  1 ! FUNCTIONAL DESCRIPTION:
;  2794     3809  1 !       This routine stores data in a fixed/variable data field.  These fields
;  2795     3810  1 !       allow a string up to 65535 bytes to be stored and retrieved by use of
;  2796     3811  1 !       extension queue records; however, a string that does not exceed the
;  2797     3812  1 !       fixed field size is stored without use of auxiliary records.
;  2798     3813  1 !
;  2799     3814  1 ! INPUT PARAMETERS:
;  2800     3815  1 !       RECORD_ADDRESS  - Pointer to record containing the fixed/variable data
;  2801     3816  1 !                         field.
;  2802     3817  1 !       FIELD_SIZE      - Size of the fixed data field.
;  2803     3818  1 !       FIELD_ADDRESS   - Address within the record of the fixed data field.
;  2804     3819  1 !       TYPE_CODE       - Value of SYM$B_TYPE for extension records.
;  2805     3820  1 !       (Length, address) pairs for each string to be stored.
;  2806     3821  1 !
;  2807     3822  1 ! IMPLICIT INPUTS:
;  2808     3823  1 !       NONE
;  2809     3824  1 !
;  2810     3825  1 ! OUTPUT PARAMETERS:
;  2811     3826  1 !       NONE
;  2812     3827  1 !
;  2813     3828  1 ! IMPLICIT OUTPUTS:
;  2814     3829  1 !       NONE
;  2815     3830  1 !
;  2816     3831  1 ! ROUTINE VALUE:
;  2817     3832  1 !       Completion status.
;  2818     3833  1 !
;  2819     3834  1 ! SIDE EFFECTS:
;  2820     3835  1 !       NONE
;  2821     3836  1 !
;  2822     3837  1 !--
;  2823     3838  1
;  2824     3839  2 BEGIN
;  2825     3840  2 MAP
;  2826     3841  2         RECORD_ADDRESS: REF BBLOCK,     ! Pointer to record
;  2827     3842  2         FIELD_ADDRESS:  REF BBLOCK;     ! Pointer to fixed/variable buffer
;  2828     3843  2 LOCAL
;  2829     3844  2         LN,                             ! Index of last non-null parameter
;  2830     3845  2         DATA_LENGTH,                    ! Total length of stored data
;  2831     3846  2         BUFFER:         BBLOCK[1024],   ! Buffer for stored data
;  2832     3847  2         CURRENT_LENGTH,                 ! Length of current string
;  2833     3848  2         CURRENT_ADDRESS;                ! Cursor for data storage area
;  2834     3849  2 BUILTIN
;  2835     3850  2         ACTUALCOUNT,
;  2836     3851  2         ACTUALPARAMETER;
;  2837     3852  2
;  2838     3853  2
;  2839     3854  2 ! Deallocate an existing variable data area, if it exists.
;  2840     3855  2 !
;  2841     3856  2 IF .FIELD_ADDRESS[FVDF_LENGTH] NEQ 0
;  2842     3857  2 THEN
;  2843     3858  2     DEALLOCATE_VARIABLE_DATA(.FIELD_SIZE, .FIELD_ADDRESS);
;  2844     3859  2
;  2845     3860  2
```

```
;  2846     3861  2 ! Strip trailing null strings from the list of string descriptors.
;  2847     3862  2 !
;  2848     3863  2 LN = 0;
;  2849     3864  2 DECR N FROM ACTUALCOUNT()-1 TO 5 BY 2 DO
;  2850     3865  3     BEGIN
;  2851     3866  3     IF ACTUALPARAMETER(.N) NEQ 0
;  2852     3867  3     THEN
;  2853     3868  4         BEGIN
;  2854     3869  4         LN = .N;
;  2855     3870  4         EXITLOOP;
;  2856     3871  3         END;
;  2857     3872  2     END;
;  2858     3873  2
;  2859     3874  2
;  2860     3875  2 ! Compute the total length of the data to be stored including the length word
;  2861     3876  2 ! for each string.
;  2862     3877  2 !
;  2863     3878  2 DATA_LENGTH = 0;
;  2864     3879  2 INCR N FROM 5 TO .LN BY 2 DO
;  2865     3880  3     BEGIN
;  2866     3881  3     DATA_LENGTH = .DATA_LENGTH + 2 + ACTUALPARAMETER(.N);
;  2867     3882  2     END;
;  2868     3883  2
;  2869     3884  2
;  2870     3885  2 ! Build a buffer containing the data to be stored.
;  2871     3886  2 !
;  2872     3887  2 CURRENT_ADDRESS = BUFFER;
;  2873     3888  2 INCR N FROM 5 TO .LN BY 2 DO
;  2874     3889  3     BEGIN
;  2875     3890  3     CURRENT_LENGTH = ACTUALPARAMETER(.N);              ! Fetch length
;  2876     3891  3     (.CURRENT_ADDRESS)<0,16> = .CURRENT_LENGTH;        ! Store length word
;  2877     3892  3     CURRENT_ADDRESS = .CURRENT_ADDRESS + 2;            ! Point past length
;  2878     3893  3     MOVC3(                                            ! Store data
;  2879     3894  3         CURRENT_LENGTH,
;  2880     3895  3         ACTUALPARAMETER(.N+1),
;  2881     3896  3         .CURRENT_ADDRESS; ..., CURRENT_ADDRESS);
;  2882     3897  2     END;
;  2883     3898  2
;  2884     3899  2
;  2885     3900  2 ! Store the data.
;  2886     3901  2 !
;  2887     3902  2 STORE_VARIABLE_DATA(
;  2888     3903  2     .RECORD_ADDRESS, .FIELD_SIZE, .FIELD_ADDRESS, .TYPE_CODE,
;  2889     3904  2     .DATA_LENGTH, BUFFER)
;  2890     3905  1 END;


                                    07FC 00000          .ENTRY  STORE_VARIABLE_DATA_LIST, Save R2,R3,R4,R5,-; 3804
                                                                R6,R7,R8,R9,R10
                                                                                                            ;
                          5E    FC00  CE  9E 00002        MOVAB   -1024(SP), SP                               ;
                          5A      0C  AC  D0 00007        MOVL    FIELD_ADDRESS, R10                          ; 3856
                                      6A  B5 0000B        TSTW    (R10)                                       ;
                                      0A  13 0000D        BEQL    1$                                          ;
                                      5A  DD 0000F        PUSHL   R10                                         ; 3858
```

```
                                08    AC   DD 00011          PUSHL   FIELD_SIZE                                                ;
                          FDB2  CF          02 FB 00014          CALLS   #2, DEALLOCATE_VARIABLE_DATA                       ;
                                          59 D4 00019 1$:        CLRL    LN                                                ; 3863
                                50        6C 9A 0001B          MOVZBL  (AP), N                                           ; 3864
                                          50 D6 0001E          INCL    N                                                 ;
                                          0A 11 00020          BRB     3$                                                ;
                                      6C40 D5 00022 2$:        TSTL    (AP)[N]                                           ; 3866
                                          05 13 00025          BEQL    3$                                                ;
                                59        50 D0 00027          MOVL    N, LN                                             ; 3869
                                          08 11 0002A          BRB     4$                                                ; 3868
                                50        02 C2 0002C 3$:      SUBL2   #2, N                                             ; 3864
                                05        50 D1 0002F          CMPL    N, #5                                             ;
                                          EE 18 00032          BGEQ    2$                                                ;
                                          57 D4 00034 4$:      CLRL    DATA_LENGTH                                       ; 3878
                                50        03 D0 00036          MOVL    #3, N                                             ; 3879
                                          09 11 00039          BRB     6$                                                ;
                                51    6C40 D0 0003B 5$:      MOVL    (AP)[N], R1                                       ; 3881
                                57    02 A147 9E 0003F         MOVAB   2(R1)[DATA_LENGTH], DATA_LENGTH                   ;
          FFF1        50        02        59 F1 00044 6$:      ACBL    LN, #2, N, 5$                                     ; 3879
                                53        6E 9E 0004A          MOVAB   BUFFER, CURRENT_ADDRESS                           ; 3887
                                56        03 D0 0004D          MOVL    #3, N                                             ; 3888
                                          10 11 00050          BRB     8$                                                ;
                                58    6C46 D0 00052 7$:      MOVL    (AP)[N], CURRENT_LENGTH                           ; 3890
                                83        58 B0 00056          MOVW    CURRENT_LENGTH, (CURRENT_ADDRESS)+                ; 3891
                                50    04 AC46 D0 00059         MOVL    4(AP)[N], R0                                      ; 3895
                          63    60        58 28 0005E          MOVC3   CURRENT_LENGTH, (R0), (CURRENT_ADDRESS)           ; 3896
          FFEA          56        02        59 F1 00062 8$:      ACBL    LN, #2, N, 7$                                     ; 3888
                                      4080  8F BB 00068          PUSHR   #^M<R7,SP>                                        ; 3904
                                        10  AC DD 0006C          PUSHL   TYPE_CODE                                         ; 3903
                                            5A DD 0006F          PUSHL   R10                                               ;
                                7E      04  AC 7D 00071          MOVQ    RECORD_ADDRESS, -(SP)                             ;
                          FEBE  CF          06 FB 00075          CALLS   #6, STORE_VARIABLE_DATA                           ;
                                            04 0007A           RET                                                       ; 3905

; Routine Size:  123 bytes,    Routine Base:  CODE + 106B
```

```
; 2892          3906  1 END
; 2893          3907  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  COMMON                         5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           4326  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                             Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;       File                           Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1       18619        55         0      1000          00:01.4


; Information:  2
; Warnings:     0
; Errors:       0


;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:QUEUEUTIL/OBJ=OBJ$:QUEUEUTIL MSRC$:QUEUEUTIL/UPDATE=(ENH$:QUEUEUTIL)

; Size:          4304 code + 5046 data bytes
; Run Time:        01:03.6
; Elapsed Time:    03:47.4
; Lines/CPU Min:    3687
; Lexemes/CPU-Min: 28146
; Memory Used:  495 pages
; Compilation Complete
```

JBCMSG
LIS

RECORDUTL
LIS

QUEUEUTIL
LIS